Construction semi-automatique d'une grammaire d'arbres adjoints pour l'analyse syntaxico-sémantique de l'arabe

Cherifa Ben Khelil

ÉCOLE DOCTORALE MATHÉMATIQUES, INFORMATIQUE, PHYSIQUE THÉORIQUE ET INGÉNIERIE DES SYSTÈMES

LIFO - Laboratoire d'Informatique Fondamentale d'Orléans / RIADI - Laboratoire de Recherche en génie logIciel, Applications distribuées, systèmes Décisionnels et Imagerie intelligente

Thèse en cotutelle internationale présentée par :

Cherifa BEN KHELIL

soutenue le : **14 Juin 2019**

pour obtenir le grade de : **Docteur de l'Université d'Orléans et de l'Université de la Manouba**

Discipline/Spécialité : **Informatique**

Construction semi-automatique d'une grammaire d'arbres adjoints pour l'analyse syntaxico-sémantique de l'arabe

Thèse dirigée par :

Chiraz BEN OHTMANE ZRIBI	Maître de Conférences, ENSI La Manouba
Denys DUCHIER	Professeur, Université d'Orléans
Yannick PARMENTIER	Maître de Conférences, Université de Lorraine

JURY :

Claire GARDENT	*Présidente du jury*	Directrice de Recherche, CNRS-LORIA UMR 7503
Kais HADDAR	*Rapporteur*	Professeur, Université de Sfax
Laura KALLMEYER	*Rapporteuse*	Professeur, Université de Düsseldorf
Chiraz BEN OHTMANE ZRIBI		Maître de Conférences, ENSI La Manouba
Denys DUCHIER		Professeur, Université d'Orléans
Yannick PARMENTIER		Maître de Conférences, Université de Lorraine

Dédicaces

Je dédie ce travail à :

Toutes les personnes qui ont contribué, de près ou de loin, à son aboutissement.
Tout chercheur et futur chercheur avide de connaissances et de savoir.

تَعَلَّمُوا الْعِلْمَ وَعَلِّمُوهُ النَّاسَ ، وَتَعَلَّمُوا لَهُ الْوَقَارَ وَالسَّكِينَةَ ، وَتَوَاضَعُوا لِمَنْ يُعَلِّمُكُمْ عِنْدَ الْعِلْمِ ، وَتَوَاضَعُوا لِمَنْ تُعَلِّمُوهُ الْعِلْمَ ، وَلا تَكُونُوا جَبَابِرَةَ الْعُلَمَاءِ ، فَلا يَقُومُ عِلْمُكُمْ بِجَهْلِكُمْ
- عُمَرَ بْنَ الْخَطَّابِ رَضِيَ اللَّهُ عَنْهُ -

"Acquire knowledge and teach people. Learn with it dignity and tranquility, humility for those who teach you and humility for those whom you teach. Do not be tyrannical scholars and thus base your knowledge upon your ignorance"
- Omar ibn al-Khattâb -

Remerciements

Tout d'abord, je tiens à remercier grandement Madame Chiraz Ben Othmane Zribi, Monsieur Denys Duchier et Monsieur Yannick Parmentier, mes directeurs de thèse, pour la confiance et les encouragements qu'ils m'ont témoignés au cours de ces années. Je leurs suis profondément reconnaissante d'avoir accepté la direction scientifique de mes travaux et de m'avoir fait bénéficier de leurs grandes compétences et leurs rigueurs intellectuelles. Soyez assuré de ma profonde gratitude.

Je remercie Monsieur Kais Haddar et Madame Laura Kallmayer d'avoir accepté de lire cette thèse et d'en être rapporteurs. Leurs lectures très attentive et leurs remarques pertinentes ont assurément contribué à améliorer la version finale de ce mémoire.

Je tiens à remercier Madame Clair Gardent d'avoir accepté d'être présidente du jury je remercie également tous les membres du jury d'avoir accepté d'évaluer ma thèse et fait le déplacement depuis l'Allemagne et la Tunisie.

Je tiens également à adresser des remerciements particuliers à Fériel Ben Fraj et Simon Petitjean pour la disponibilité et l'immense aide pour mener à bien ce travail.

Je tiens à exprimer ma reconnaissance aux membres du laboratoire LIFO pour l'accueil, la sympathie et la bienveillance.

Merci aux membres de l'équipe « grammaire formelle » qui ont répondu à plusieurs de mes préoccupations, ce qui m'a permis de progresser.

J'ai également eu la chance d'intégrer le programme SToRE de Université Heinrich Heine de Düsseldorf. Ce stage a contribué de manière significative à mon avancement, c'est une expérience très intéressante, je tiens à remercier tous les membres de ce programme et particulièrement Eva et mes collègues de bureau Rafael et Ben.

Je tiens à remercier Nabil et Isabelle pour leur sympathie et leur efficacité dans l'organisation et la résolution des problèmes administratifs.

Merci à tous les docteurs et doctorants que j'ai pu côtoyer, merci à Fatma, Asma, Asma et Asma, Binh, Lynh, Tendrie, Yosra et Olfa, d'Orléans, de la Manouba et d'ailleurs, merci de votre soutien et votre écoute, tous les moments que j'ai partagé avec vous étaient des moments de pur bonheur.

Merci à toutes les personnes qui de près ou de loin m'ont aidés et encouragés.

J'exprime ma gratitude à tous mes ami(e)s qui se reconnaitront sûrement ainsi que mes anciens étudiants, aux membres de ma famille et tout particulièrement Ramzi, Dorsaf et leurs enfants et la famille de Majdi Mouha.

Enfin, ma profonde reconnaissance à mon soutien affectif le plus précieux, Papa Fethi, Maman Fatma vous êtes mes héros, ma sœur Meriem, mon frère Mohamed Amine, mon beau-frère Bassem, mon petit neveu trop mignon Omar, tontons, tatas cousins et cousines vous êtes mes amours.

Sommaire

Liste des figures

Liste des tableaux

Introduction générale

Une grammaire formelle est un moyen de représentation permettant de définir une syntaxe ou plus largement un langage formel. Cette notion est particulièrement utilisée en compilation des langages de programmation, dans la théorie de la calculabilité et dans le domaine du Traitement Automatique du Langage Naturel (TALN). Dans le contexte du TALN, il y a eu chronologiquement, plusieurs générations de grammaires, dont les grammaires d'unification. Ce courant de grammaire a été développé à la fin des années 70 en réaction au courant transformationnel de Chomsky afin de corriger certaines de ses limites et ses insuffisances à l'implémentation. Les grammaires d'unification sont basées sur les structures de traits et mettent au premier plan, l'interfaçage de la syntaxe avec le lexique et la sémantique. Cet interfaçage est primordial pour relier la signification d'une phrase à sa structure syntaxique.
En effet, la construction automatique du sens d'une phrase représente un défi de taille pour de nombreuses applications dans le domaine du TALN telles que les systèmes de traduction automatique, de dialogue homme-machine et de questions-réponses. Cependant, pour construire une représentation de la signification d'une phrase (ou d'une déclaration), il est généralement essentiel de produire une représentation de sa structure syntaxique. Cette relation peut être établie à l'aide d'une interface syntaxe-sémantique. Cette dernière permet la construction progressive de la représentation sémantique de la phrase parallèlement à celle de sa structure syntaxique. Cela implique que l'interface syntaxe-sémantique est étroitement liée à la grammaire ou au formalisme grammatical choisi.
Disposer de ressources électroniques telles que des grammaires est très utile, voire même indispensable pour le traitement d'une langue naturelle. Ceci est spécialement vrai pour la langue arabe qui est une langue difficile en matière de production grammaticale. En effet, à ce jour, il n'existe pas de grammaire formelle à couverture étendue pour la langue arabe intégrant la dimension sémantique. Et pour cause, cette langue sémitique présente beaucoup de spécificités telles que l'ordre de mots relativement libre, combiné à une morphologie riche et l'omission de diacritiques (voyelles) dans les textes écrits. Bien que plusieurs travaux de recherche aient abordé certaines de ces problématiques, les ressources numériques utiles pour le traitement de l'arabe demeurent relativement rares ou encore peu disponibles.
C'est dans ce cadre que s'inscrit notre travail de thèse qui vise à élaborer une grammaire formelle à dimension sémantique pour l'utiliser dans les applications du traitement automatique de la langue arabe. Notre choix s'est porté sur l'un des formalismes des grammaires d'unifications à savoir les grammaires d'arbres adjoints (Tree-adjoining grammar : TAG) [Joshi et al., 1975]. Ce formalisme possède un pouvoir riche de représentation. Il utilise la notion des structures arborescentes dites arbres élémentaires pour la représentation des éléments syntaxiques (structures simples, complexes, combinatoires, partagées). Ces structures peuvent être combinées au moyen d'opérations spécifiques pour construire l'arbre syntaxique complet d'une phrase. De plus, TAG a connu plusieurs extensions afin de satisfaire les exigences de ses utilisateurs et aussi de l'accommoder aux besoins de représentation des différentes langues traitées telles que l'anglais, le français, l'allemand et aussi l'arabe.
La construction d'une grammaire TAG à portée sémantique est possible puisque ce formalisme accepte l'intégration de ce type d'informations. Il existe divers formats de représentation de cette information sémantique. Notre intérêt s'est orienté vers un format très adaptée à la tâche de composition sémantique à savoir les cadres sémantiques [Fillmore, 1982].

Ces derniers offrent une représentation riche et hiérarchiquement structurée de la sémantique.
Ce travail de thèse que nous proposons prend ses origines du travail réalisé dans le cadre de la thèse de [Ben Fraj, 2010] effectuée au sein du laboratoire RIADI-GDLRIADI [1]. Cette thèse avait abouti, entre autres, à la construction d'une grammaire TAG pour l'arabe baptisée ArabTAG, utilisée pour la construction d'un corpus arboré (Treebank).
Contrairement à cette grammaire construite manuellement, nous proposons une nouvelle version plus riche générée semi automatiquement. Les spécificités de l'arabe et la difficulté de conception d'une grammaire en termes de coût et de temps nous ont motivés à utiliser des langages de description extrêmement expressifs tels que les langages de description méta-grammaticales. De plus, nous étendons davantage cette description en rajoutant une dimension sémantique.
Somme toute, nos objectifs fondamentaux se résument comme suit :

— Décrire la syntaxe et la sémantique de l'arabe au moyen d'une méta-grammaire,

— Générer semi-automatiquement une grammaire TAG à portée sémantique pour l'arabe.

La grammaire que nous proposons a été produite semi-automatiquement au moyen du formalisme XMG (eXtensible MetaGrammar) [Crabbé et al., 2013]. À partir d'une représentation abstraite de la syntaxe (la description méta-grammaticale ArabicXMG) nous avons généré ArabTAG V2.0. Ensuite, nous avons étendu cette grammaire en intégrant des informations sémantiques (la sémantique des cadres). Enfin, afin d'évaluer cette grammaire, nous avons mis en place un processus d'analyse syntaxico-sémantique en utilisant un corpus issu d'extraits d'un manuel scolaire d'apprentissage de l'arabe.
Ce manuscrit est composé de cinq chapitres scindés en trois grandes parties : L'état de l'art, la contribution et l'évaluation. L'étude de l'art comprend les trois premiers chapitres. Ces derniers détaillent les points suivants :

— Le Chapitre 1 présente les différents niveaux d'analyse d'une langue naturelle et s'intéresse particulièrement aux niveaux morphologiques, syntaxiques et sémantiques et le mécanisme d'interfaçage entre eux. Une étude des principaux formalismes de représentation syntaxique et sémantique est réalisée. L'objectif de cette étude est d'explorer les avantages et les limites de ces systèmes de représentation. Le résultat de cette analyse nous a guidé dans le choix des formalismes utilisés dans notre approche.

— Le Chapitre 2 est consacré à l'étude de la langue arabe dans sa version standard moderne. Il se focalise sur les spécificités morpho-syntaxiques de cette langue qui la rendent relativement difficile à traiter. Ce chapitre présente aussi un ensemble de ressources numériques utiles pour le traitement de l'arabe ainsi que des travaux antérieurs réalisés pour la construction d'une TAG pour l'arabe. Nous nous sommes particulièrement intéressés aux insuffisances de chacune de ces approches.

— Le Chapitre 3 conclut la partie état de l'art et marque la transition vers la partie contribution en introduisant un formalisme primordial dans le développement de note approche. L'étude présentée au sein de ce chapitre met en avant les motivations et l'importance de l'utilisation des langages de description dans la construction

1. Laboratoire de Recherche en génie logIciel, Applications distribuées, systèmes Décisionnels et Imagerie intelligente

des grammaires TAG. Nous nous sommes particulièrement intéressés à l'un de ces systèmes de description à savoir XMG ainsi que sa version étendue XMG2. La deuxième partie de ce chapitre expose les fonctionnalités dont XMG dispose et qui nous ont paru profitables pour la construction de notre TAG pour la langue arabe.

La partie contribution est présentée dans le Chapitre 4. Elle est initiée par notre approche de description de la syntaxe de l'arabe au moyen d'une méta-grammaire suivie d'un parcours des différents phénomènes linguistiques couverts par la grammaire générée ArabTAG V2. La deuxième partie de ce chapitre détaille le processus d'intégration des informations sémantiques au sein de la méta-grammaire.
Le Chapitre 5 décrit le processus de l'évaluation de la grammaire générée. Cette évaluation est divisée en deux phases : l'évaluation syntaxique de la couverture de la grammaire suivie d'une analyse syntaxico-sémantique. Ainsi, le protocole de chacune de ces évaluations est présenté en décrivant : le corpus de test utilisé, les étapes de construction du lexique, le processus d'analyse (syntaxique et syntaxico-sémantique) et enfin les résultats obtenus. Ces derniers sont analysés et exploités afin de cerner la pertinence de notre approche et déceler ses limites.
Enfin, nous concluons ce mémoire de thèse avec un récapitulatif des apports de notre travail, les questions qu'il soulève et les perspectives.

Première partie

État de l'art

Chapitre 1

Les formalismes dédiés au Traitement Automatique des Langues

1.1 Introduction

Avec l'avènement de l'intelligence artificielle, la capacité à comprendre et à traiter les langues naturelles par un ordinateur est devenue un besoin nécessaire pour mener à bien des activités importantes telles que : la traduction automatique, l'analyse et l'interprétation de texte, la détection des erreurs dans un texte, etc. Ces systèmes et applications de Traitement Automatique de la Langue Naturelle (TALN) doivent être dotés d'un mécanisme de compréhension et/ ou traitement des énoncés linguistiques. Ceci est traduit par une suite de sous-traitements des différents niveaux de description (phonologique, morphologique, syntaxique, sémantique et pragmatique) d'une langue naturelle.
Les efforts de recherche se sont succédés pour offrir des outils d'analyse de plus en plus performants pour automatiser cette chaîne de traitement. Plus la tâche de chaque niveau est correctement accomplie, plus les performances des systèmes qui l'utilisent sont élevées. Néanmoins, pour certaines applications de TALN, il n'est pas indispensable de passer par tous les niveaux d'analyse. Tout de même, l'interaction entre ces niveaux demeure primordiale. Parmi les moyens d'interaction possibles, l'utilisation des interfaces pour établir un lien entre deux niveaux d'analyse.
Dans cette thèse, nous nous intéressons à l'analyse syntaxico-sémantique. Il convient donc de proposer, dans ce qui suit, un tour d'horizon des principes et formalismes utilisés dans le domaine du TALN pour l'analyse syntaxique et sémantique. Le présent chapitre est subdivisé en trois grandes sections : la section 1 donne un aperçu général sur le processus d'analyse d'une langue naturelle et s'intéresse particulièrement aux niveaux syntaxique et sémantique et leur interfaçage. La section 2 est consacrée à la présentation d'un ensemble de formalismes des grammaires d'unification pour décrire la syntaxe d'une langue donnée. Tandis que la dernière section expose un panorama des différents formalismes de représentations sémantiques.

1.2 Différents niveaux d'analyse d'une langue naturelle

Le traitement d'un texte par les systèmes de TALN suit généralement un processus d'analyse décomposé en cinq étapes :

1. Traitement phonétique : permet d'étudier les sons des langues naturelles, indépendamment de leur sens. Cette étape est opérée dans le cas où l'entrée du système est vocale. À partir de cette entrée, les informations linguistiques sont extraites : les phonèmes [1] et la prosodie. [2]
2. Traitement morphologique : permet d'étudier la formation des mots à partir d'unités plus petites appelées morphèmes. Cette étape affecte les informations grammaticales à chaque mot d'un texte indépendamment du contexte.
3. Traitement syntaxique : permet d'analyser la structure de la phrase en déterminant les relations grammaticales entre les mots ou groupes de mots.
4. Traitement sémantique : permet d'étudier le sens qui peut être associé à un mot ou à une phrase. La construction d'une représentation de ce sens est réalisée en faisant correspondre à chaque structure, reconnue par le niveau syntaxique, des objets, des actions ou des situations dans un monde de référence (réel ou imaginaire).
5. Traitement pragmatique : est un courant dans l'étude du discours. [3] Il permet d'interpréter la phrase en fonction du contexte de son emploi. Ce dernier est représenté par l'ensemble des connaissances linguistiques présentes autour des unités lexicales de la phrase. Par exemple la situation de communication, le cadre spatio-temporel, l'âge, le genre des locuteur(s), etc.

Ces différents niveaux peuvent interagir séquentiellement ou parallèlement entre eux durant l'analyse d'un texte. Néanmoins, les systèmes de TALN ne sont pas contraints de passer par les cinq étapes de ce processus. Par exemple, les systèmes de reconnaissance optique de caractères, les claviers auto-correcteur, les correcteurs d'orthographe ou de syntaxe, etc. ne traitent pas des niveaux sémantique et pragmatique. Dans notre thèse, nous nous intéressons au traitement de la langue écrite, qui ne nécessite pas le passage par le traitement phonétique. Ainsi, dans ce chapitre, nous présentons les trois niveaux suivants : morphologique, syntaxique et sémantique.
Nous nous focalisons plus particulièrement sur les deux derniers niveaux syntaxique et sémantique objets de notre étude.

1.2.1 Traitement morphologique

La morphologie étudie la structure interne et externe des mots. Elle utilise des règles lexicales pour délimiter les morphèmes [4] d'un mot et les étiqueter par les informations nécessaires qui les qualifient. Durant ce processus, chaque mot de l'énoncé voit toutes ses analyses morphologiques possibles déterminées. Chacune de ces analyses inclut un seul

1. Les phonèmes sont les sons successifs qui constituent le mot.
2. La prosodie c'est l'intonation et rythme du son permettant de spécifier par exemple l'interrogation, l'injonction, l'exclamation, etc.
3. Selon Anne-Marie Diller et François Récanati la pragmatique étudie l'utilisation du langage dans le discours, et les marques spécifiques qui, dans la langue, attestent sa vocation discursive.
4. Unité linguistique minimale, non décomposable, porteuse de sens.

choix de partie du discours (POS, Part-Of-Speech), appelée aussi classe grammaticale ou catégorie grammaticale (exemples : nom, verbe, adjectif, pronom, préposition, conjonction, etc.). Un mot peut avoir différentes catégories grammaticales. Dans ce cas, on parle d'ambiguïté grammaticale et il faudra passer par l'étape de "la désambiguïsation grammaticale".
La résolution de cette ambiguïté peut être assurée par un étiquetage morphosyntaxique, aussi appelé étiquetage grammatical, ou encore POS tagging (Part-Of-Speech tagging). Ce processus consiste à associer au mot l'ensemble de ses traits morphologiques (ou morphosyntaxiques) correspondants tels que le cas, le genre, le nombre ainsi que le lemme. [5]
Aujourd'hui, l'analyse morphologique ou morphosyntaxique des mots dispose d'un ensemble efficace de programmes permettant d'obtenir des résultats fiables pour différentes langues naturelles, notament pour la langue arabe.

1.2.2 Traitement syntaxique

Alors que la morphologie s'intéresse à la structure des mots, la syntaxe décrit la manière dont les mots sont organisés pour former des syntagmes ou des phrases. L'automatisation de la tâche d'analyse syntaxique requiert un ensemble de connaissances et de ressources fournissant les informations sur les structures correctes des données fournies à l'entrée (texte ou phrase). En d'autres termes, le processus d'analyse doit être conforme aux règles d'une grammaire formelle ou disposer d'une méthode d'acquisition des connaissances syntaxiques à partir d'une ressource linguistique telle qu'un corpus arboré (Treebank) [Ratnaparkhi et al., 1994].
Les corpus arborés définissent un grand ensemble de données textuelles stockées sous forme de phrases analysées et annotées d'arbres syntaxiques. Ces données annotées sont très utiles dans la création d'analyseurs syntaxiques robustes à base d'apprentissage. En effet, l'analyseur syntaxique constitue sa base de connaissance à partir des données extraites du corpus arboré. Ainsi, analyser une phrase revient à attribuer l'arbre approprié parmi les arbres qui sont stockés dans cette base de connaissance. Ceci nous ramène au problème majeur des approches à base de corpus. En effet, l'analyse est limitée aux structures syntaxiques définies dans ces corpus. De plus, la disponibilité de ce type de ressources lexicales est très contrastée selon la langue. En conséquence, il est difficile de réaliser des résultats significatifs en appliquant cette méthode pour les langues peu dotées de ressources linguistiques.
D'autres méthodes d'analyse ont adopté l'approche basée sur des règles, qui utilise des grammaires formelles bien définies pour représenter la syntaxe. Contrairement aux approches à base de corpus, les grammaires construites sont capables de couvrir un ensemble plus important de structures syntaxiques d'une langue. Néanmoins, la mise en place de telles grammaires nécessite beaucoup d'effort et peuvent être très couteuse en temps. Nous avons étudié l'évolution de ces différentes théories utilisées en linguistiques informatiques à savoir : les grammaires génératives, les grammaires transformationnelles et les grammaires d'unification.

5. La forme canonique du mot.

1.2.2.1 Générations de grammaires formelles

Dans cette section, nous présentons les différentes générations de grammaires formelles suivant l'ordre chronologique de leur apparition.

1.2.2.1.a Grammaire générative

Vers la fin des années 1950, le linguiste américain Noam Chomsky introduit la notion de grammaire générative. Avec cette notion, Chomsky soulève le problème de l'apprentissage de la langue par l'enfant et considère qu'une théorie linguistique doit s'occuper de l'aspect créateur du langage. C'est l'un des principes fondamentaux de sa grammaire générative qui distingue la capacité langagière de l'acte de parole.
En effet, tout locuteur natif possède une connaissance innée des mécanismes du langage. Il peut distinguer une phrase grammaticale d'une phrase agrammaticale. Cette attitude est appelée compétence linguistique. Lorsque les individus utilisent ces aptitudes dans des actes de parole, on parle du modèle de la performance [Ruwet, 1968]. En d'autres termes, ce que Chomsky propose est une distinction entre le modèle de compétence et le modèle de performance, fondée sur l'hypothèse de l' "innéisme" des capacités linguistiques [Chomsky, 1980].
La théorie générative a établi une hiérarchie entre classes de langages ou types de grammaires, connue sous le nom de la hiérarchie de Chomsky. Cette hiérarchie a été créée en fonction de la nature des règles de production. En apportant des restrictions à la forme des règles de production de la grammaire, la classe des langages qui peuvent en être engendrés peut-être restreinte. Ainsi quatre catégories de grammaires ont été définies :

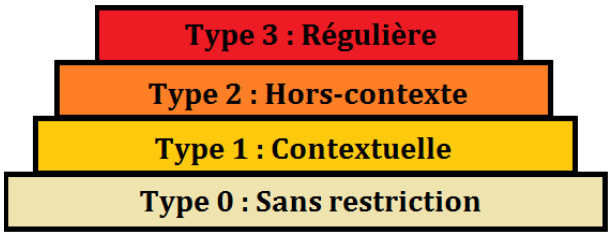


FIGURE 1.1: La Hiérarchie de Chomsky

Les grammaires de type 0 (en anglais unrestricted grammars) ne connaissent aucune restriction. Elles engendrent des langages récursivement énumérables qui sont semi-décidables. [6]
Les grammaires contextuelles, appelées aussi sensibles au contexte (en anglais context-sensitive grammar), constituent le type 1. Les règles de production de ce type de grammaires sont de la forme : uXv → uwv Où u, v et w sont des terminaux quelconques (avec w non vide) et X est un symbole non terminal. Ainsi, le remplacement de X par w se fait en présence du "contexte" (u,v). Le langage engendré est donc récursif et reconnaissable par un automate borné linéairement.
Ces grammaires peuvent servir à déterminer si un mot est approprié ou non dans un

6. Un langage est décidable si pour tout mot ou toute phrase on peut savoir au bout d'un temps fini si elle appartient ou non au langage. Sinon il est dit indécidable.

certain contexte. Une telle dépendance au contexte n'est pas autorisée pour les grammaires type2, appelées grammaires hors-contexte (en anglais context-free grammars). Les parties gauches de ses règles de production contiennent un unique non-terminal, donc sa dérivation ne dépend d'aucun contexte : X → w avec X est non-terminal et w une chaîne quelconque du vocabulaire de la grammaire.
Cette grammaire permet notamment de définir les langages algébriques qui sont le fondement théorique de la syntaxe de la plupart des langages de programmation. Ces langages sont reconnaissables par un automate à pile non déterministe. Enfin, les grammaires de type 3 sont dites grammaires régulières (en anglais regular grammar). Elles sont divisées en deux types :

— La grammaire linéaire à droite est une grammaire dont chaque membre droit de la règle finit par un non-terminal : X → a, X → aY : le membre à gauche X est un unique non-terminal et le membre à droite est soit une chaîne de terminaux a, soit une chaîne de terminaux suivie par un seul non-terminal Y.

— La grammaire linéaire à gauche est une grammaire dont chaque membre droit de la règle commence par un non-terminal : X → a, X → Ya : le membre à gauche X est un unique non-terminal et le membre à droite est soit une chaîne de terminaux a, soit un non-terminal Y suivi par une chaîne de terminaux.

Les langages engendrés correspondent aux langages rationnels qui sont reconnus par des automates à nombre fini d'états.
Le modèle de la grammaire générative de Chomsky est limité par son pouvoir représentationnel faible de la langue naturelle. Il est incapable de traiter les spécificités de la langue naturelle telles que l'enchâssement et l'ordre des mots semi-contraints dans certaines langues (négation en français, verbe en 2ième position en allemand, etc.). Chomsky ne tarde pas à proposer une solution avec l'ajout de nouvelles règles appelées : règles de transformation. D'où l'apparition des grammaires transformationnelles.

1.2.2.1.b Grammaire transformationnelle

La grammaire transformationnelle, introduite par Chomsky au début des années 60, décrit une langue comme une série de dérivations et de transformations à partir des phrases de base. Ces transformations sont effectuées par diverses opérations telles que le déplacement de lexèmes ou de morphèmes, la permutation, l'enchâssement ou encore la substitution.
Avec la grammaire transformationnelle, Chomsky propose une organisation en une structure profonde syntagmatique, une structure de surface et un composant transformationnel [Notari, 2010] :

— Les structures profondes (en anglais : deep structure) sont les phrases (une phrase est réécrite en un syntagme nominal plus un syntagme verbal : P → SN+ SV) produites inconsciemment par le locuteur. C'est à ce niveau où figure tout ce qui est nécessaire à l'interprétation sémantique.

— Les structures de surface (en anglais : "surface structure") sont les résultats des opérations complexes ou transformations à partir de la structure profonde.

— Le composant transformationnel permet de convertir une structure profonde en une de surface.

Soit la structure profonde suivante qui correspond à la phrase : "le conducteur a provoqué un accident".
Après différentes transformations de la phrase de départ on peut obtenir les structures de surface suivantes :

1. *Le conducteur imprudent a provoqué un accident* : La transformation consiste à l'ajout de l'adjectif qualificatif "imprudent".

2. *Le conducteur qui était imprudent a provoqué un accident* : La transformation continue avec l'addition d'une subordonnée relative (pronom relatif *qui* et le verbe conjugué *était*).

3. *L'accident a été provoqué par le conducteur imprudent* : Cette transformation est définie par une succession d'opérations : permutation des syntagmes nominaux, addition de l'adjectif qualificatif "imprudent" à la fin et mise en forme passive du verbe "provoquer".

En d'autres termes, une grammaire transformationnelle rend compte de la grande variété des transformations grammaticales que l'on peut appliquer afin de passer de la structure profonde à la structure de surface, comme par exemple les transformations : passive, négative, interrogative, déclarative, etc.
Cependant, l'ordre de ces transformations est parfois difficile à contrôler. De plus, l'application de ce genre de grammaires est très complexe en raison du nombre important de ses règles de production et ses règles transformationnelles. Dans le prolongement de la grammaire transformationnelle de Chomsky ou plus exactement en réaction à celle-ci, certains linguistes [Yngve, 1960] [Harman, 1963] ont proposé de conserver les grammaires syntagmatiques en accroissant leur capacité descriptive. Leur idée était d'enrichir les grammaires syntagmatiques de telle sorte qu'elles permettent l'expression claire et distincte des principes organisateurs des langues. Ainsi, une nouvelle génération de grammaire est apparue. Il s'agit de la grammaire d'unification.

1.2.2.1.c Grammaire d'unification

Les grammaires d'unification ont été développées à la fin des années 70 en réaction au courant transformationnel de Chomsky afin de corriger certaines de ses limites formelles et ses insuffisances à l'implémentation. Elles mettent, au premier plan, l'interface de la syntaxe avec le lexique et la sémantique et se présentent aujourd'hui comme une alternative au modèle générativiste dominant. Les grammaires d'unification ont introduit la notion de traits dans les règles de réécriture. Lesdites structures sont un ensemble de couples attribut-valeur, les valeurs pouvant être des symboles atomiques ou des traits. Elles fournissent des informations partielles qui peuvent être enrichies.
La figure 1.2 illustre la structure de traits relative au nom "grammaire". Elle énumère les caractéristiques de ce mot sous la forme d'une liste structurée. La forme "grammaire" est décrit comme un nom féminin singulier. De plus, les deux traits "animé" et "humain" permettent d'apporter quelques informations sémantiques sur ce mot.

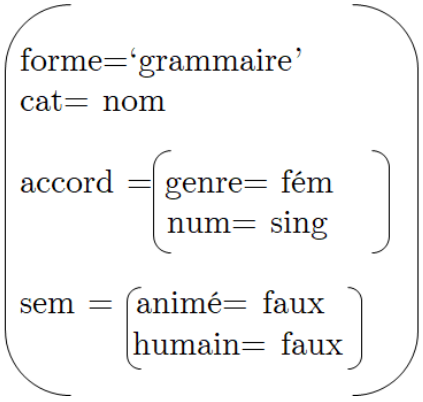


FIGURE 1.2: La structure de traits relative au nom "grammaire"

L'avantage dans l'utilisation des structures des traits est la possibilité de regrouper dans une seule structure des informations de différentes natures qu'elles soient morphologiques, syntaxiques ou sémantiques et aussi d'exprimer des corrélations éventuelles. Ces structures permettent d'affiner les contraintes au niveau du lexique et ainsi de simplifier les règles de grammaire.
L'unification de structures de traits est une opération proche de l'union ensembliste. Elle représente l'un des moyens les plus efficaces et les plus puissants pour traiter les informations linguistiques. L'unification intervient lorsqu'on veut enrichir la description d'un objet en combinant les informations de deux structures de traits (A et B) qui le décrivent. La structure résultante de l'unification représente la structure de traits minimale qui est à la fois extension de A et de B. En revanche, si les deux structures portent des informations incompatibles, l'unification échoue.
La figure 1.3, présente trois structures de traits : [7] (1), (2) et (3). L'unification est opérée au niveau du trait "accord". Les structures de traits (1) et (2) sont unifiées puisque les valeurs du "num" de l'accord de (1) et (2) sont identiques à savoir plur (pluriel). Ainsi, une nouvelle structure de traits (4) est produite. En revanche, l'unification échoue pour (1) et (3), car le chemin <accord num> contient des valeurs incompatibles : sing (singulier) et plur (pluriel).

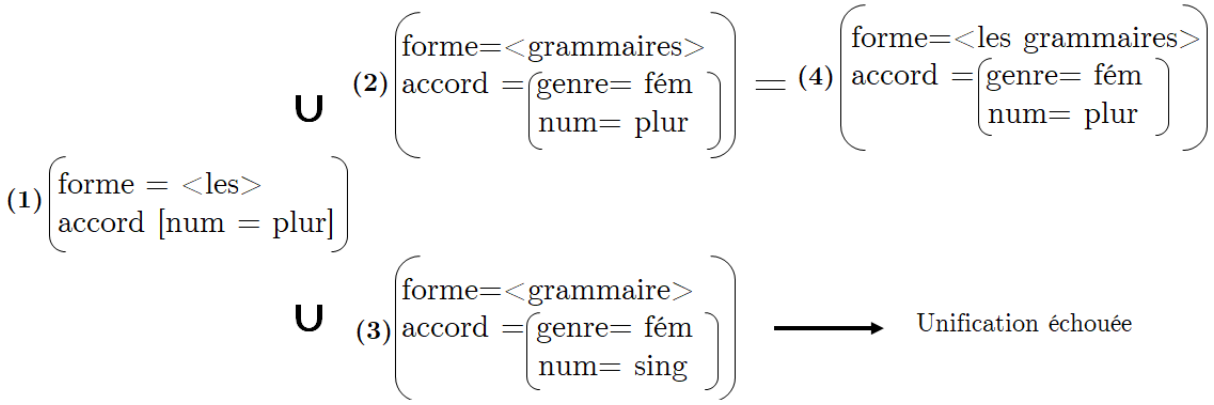


FIGURE 1.3: Exemple d'unification des structures de traits

7. Il s'agit de structure non typée

L'unification est un système formel permettant de gérer efficacement des catégories syntaxiques riches. Elle présente une approche très intéressante pour le traitement de la langue naturelle. Hormis son indépendance de l'ordre, elle permet de combiner les informations associées aux mots et aux syntagmes pour construire celles de la phrase, et aussi de vérifier leur compatibilité.

1.2.2.2 Discussion

Nous avons présenté, dans cette section, trois générations de grammaires (figure 1.4). La première, est la grammaire générative de Chomsky qui a révolutionné le domaine de la linguistique et du traitement des langues naturelles. Ce modèle fut critiqué par sa rigidité et son manque de précision formelle.
En effet, Chomsky voulait que la grammaire exprime les propriétés communes à plusieurs types d'entités syntaxiques (exemple : l'enchâssement, la coordination et la subordination, etc.). Cependant, le modèle de grammaires qu'il a présenté était limité, incapable de traiter ces spécificités de la langue naturelle. De plus, de nombreux linguistes, tels que Jean Piaget [Piattelli-Palmarini, 1980] et Riny Huybregts [Van Riemsdijk, 1982], ont considéré que cette théorie est abstraite et n'est pas suffisante pour représenter toutes les contraintes syntaxiques d'une langue naturelle. Conscient de ces limites, Chomsky a proposé l'ajout de nouvelles règles appelées règles de transformation. Ce modèle de grammaire transformationnelle a permis de traiter les traits manquants de la grammaire générative. Néanmoins, ce procédé a entrainé l'explosion du nombre des règles de production et celles transformationnelles. Par conséquence, l'application de ce genre de grammaires devient très complexe.
En outre, les grammaires transformationnelles mettent l'accent sur les informations syntaxiques sans prendre en considération la sémantique malgré son importance dans la représentation syntaxique des langues naturelles.
Le courant de grammaires d'unification a apporté une solution pour la représentation sémantique en introduisant la notion des structures de traits. Ces dernières visent à regrouper les différentes informations morphologique, syntaxique et sémantique dans une même représentation.

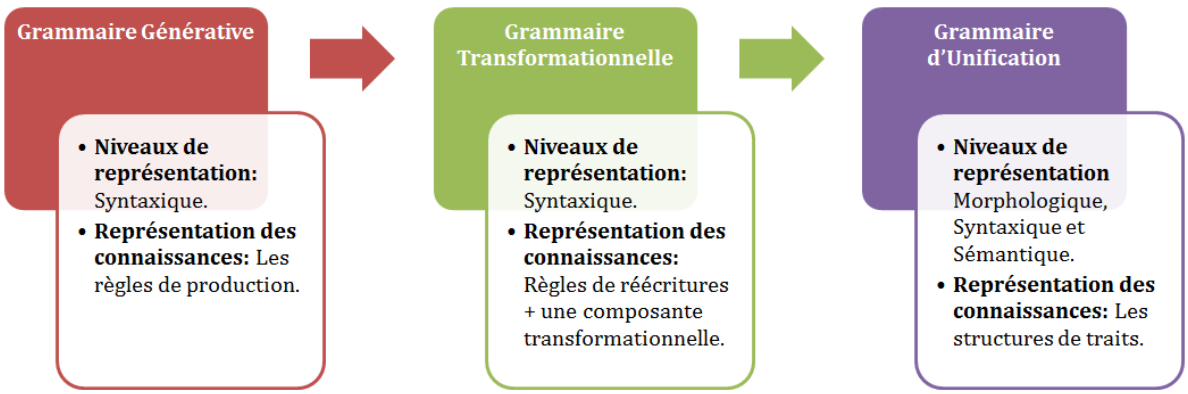


FIGURE 1.4: L'évolution des grammaires

Vu les insuffisances des premières générations de grammaires, nous avons choisi de

nous focaliser sur les formalismes de la troisième génération des grammaires d'unification dans la suite de nos travaux. Ces formalismes seront présentés dans la section 2 de ce chapitre.

1.2.3 Traitement sémantique

L'analyse sémantique est l'étude du sens d'un mot ou d'une phrase. Elle est utile pour diverses applications du domaine TALN tels que les systèmes de questions/réponses, la traduction automatique, la recherche des documents etc. En informatique, la description du sens d'un mot ou une phrase peut être réalisée en définissant un ensemble règles de calcul de sa dénotation. Il existe deux types d'approches d'étude de la sémantique : la sémantique formelle et la sémantique lexicale.
La sémantique lexicale est l'étude du sens des mots (morphèmes) et les différentes relations établies entre eux. Elle s'intéresse ainsi à l'organisation sémantique du lexique qui regroupe les relations hiérarchiques et d'inclusion ou encore les relations d'équivalence et d'opposition. Parmi ces relations, nous pouvons citer la méronymie, la synonymie , l'hyperonymie ou encore l'hyponymie.
Dans le cadre de la sémantique lexicale, l'analyse du sens d'un mot est effectuée à partir d'un :

— Dictionnaire : La similarité de deux mots est définie en fonction de définitions. En d'autres termes, plus leurs définitions vont avoir de vocabulaire commun, plus ces deux mots sont similaires.

— Thésaurus : C'est une ressource lexicale possédant des liens (en particulier liens d'hyperhonymie) entre ses entrées. Par exemple, la similarité entre les mots est calculée à partir de la position des nœuds dans le thésaurus.

— Corpus : l'analyse repose sur des méthodes sémantiques distributionnelles. Ces dernières calculent la proximité sémantique entre mots sur la base des contextes qu'ils partagent dans un corpus donné.

Comme toute discipline du TALN, la sémantique lexicale a besoin de méthodes et de formalismes de représentations. Les principaux cadres formels utilisés dans ce type d'approche sont les graphes, la logique mathématique (le lambda-calcul) et les espaces vectoriels. Dans le cadre de notre travail, notre intérêt se porte plutôt sur l'analyse sémantique en faisant intervenir la syntaxe. Parmi les méthodes couramment employées en sémantique, l'utilisation d'un langage symbolique et formel dans lequel le sens des expressions, d'une langue donnée, est décrit de manière concise et précise. Dans la sémantique formelle, ceci se traduit par une représentation sémantique logique, interprétable dans un modèle. La construction d'un tel modèle pour une phrase repose sur deux éléments : la sémantique des mots qui y figurent et sa structure syntaxique.
À l'aide des informations obtenues par l'analyse syntaxique, le sens de la phrase peut être déduit grâce à la connaissance des relations existantes entre les mots. En d'autres termes, il est indispensable de disposer d'un formalisme grammatical qui offre une description des liens syntaxiques entre les constituants de la phrase. Cette structure syntaxique (par exemple un arbre résultat de l'analyse) sert de base aux mécanismes compositionnels de la sémantique.
Les travaux de Richard Montague [Montague, 1974b], [Dowty et al., 1981] ont permis

l'avènement de la sémantique formelle. Ces travaux s'inspirent de ceux de de Frege sur la sémantique des langages formels et le principe de compositionnalité. Ils furent par la suite la base de plusieurs approches de formalismes de représentation et de calcul sémantique. En résumé, dans la sémantique formelle la signification d'une expression dépend de la composition et la signification de ses parties. Alors que la sémantique lexicale tend plutôt à analyser un texte en montrant comment ses mots donnent du sens en fonction des types des contextes où ils apparaissent. Nous avons donc choisi d'approfondir notre étude sur les formalismes de représentation sémantique au sein de la sémantique formelle. Mais avant d'entamer cette étude, il est primordial de savoir comment interfacer les niveaux : morphologique, syntaxique et sémantique.

1.2.4 Interface entre les niveaux de traitement

La notion d'interface permet d'entretenir les relations entre les différents niveaux traitant de la morphologie, de la syntaxe et de la sémantique au sein d'une grammaire formelle. Nous nous focalisons dans ce qui suit à l'interface morphologie-syntaxe et l'interface syntaxe-sémantique.

1.2.4.1 Interface morphologie-syntaxe

Afin de rendre compte de certains phénomènes de la langue, il est primordial de faire interagir la morphologie et la syntaxe. Par exemple, la morphologie intervient dans l'accord ou le désaccord entre les constituants d'une phrase. Prenons l'exemple de la phrase suivante : "La sœur de Ali, inquiète, est partie à l'hôpital". Cette phrase est syntaxiquement correcte. Le genre et le nombre de l'adjectif "inquiète" indique avec quel mot ou groupe de mot s'établit le rapport morphosyntaxique qui est "la sœur de Ali". En revanche, dans la phrase "La sœur de Ali, inquiètes, est partie à l'hôpital" l'accord en nombre de l'adjectif n'est pas respecté. Cette phrase devient ainsi syntaxiquement incorrecte.
L'interface morphologie-syntaxe représente un trait d'union entre les variations morphologiques et le contexte syntaxique. Elle peut être représentée au moyen des structures de traits ou encore par un ensemble de règles comme par exemple dans les Grammaires à clauses définies (DCG : Definite Clause Grammar) introduites par [Pereira and Warren, 1980]. Ce type de grammaires permet de traduire les règles de réécriture (indépendantes du contexte) directement dans une forme exécutable par Prolog.

1.2.4.2 Interface syntaxe-sémantique

La construction compositionnelle du sens des phrases à partir de leur analyse syntaxique nécessite de mettre en place un format de notation qui permet de lier les analyses de la syntaxe et de la sémantique.
Dans une grammaire formelle, le lien entre ces deux niveaux peut être établi en utilisant une interface syntaxe-sémantique. Cette dernière se charge de superviser la construction du sens de la phrase en unifiant les informations sémantiques de ses constituants. L'enjeu est donc de savoir comment développer et explorer un système de règles qui assurent l'interface entre la syntaxe et la sémantique.

Afin d'accomplir cette tâche, il est nécessaire de bien choisir le formalisme de représentation des structures syntaxiques permettent d'intégrer une telle interface. Grâce à l'héritage de Montague [Montague, 1970], qui propose d'associer à chaque entrée lexicale (ou règle syntaxique) de la grammaire une représentation sémantique (respectivement une règle de calcul sémantique) ce procédé est réalisable dans le cas des grammaires hors contexte. En effet, la définition d'une interface syntaxe-sémantique est possible au sein des grammaires d'unification au moyen de variables d'unification. L'idée est de définir, tout d'abord, les représentations sémantiques (se référer à la section 3 de ce chapitre) sous forme de structures de traits, ensuite mettre en place une interface syntaxe-sémantique permettant aux variables d'unification contenues dans ces structures d'être unifiées correctement lors de la composition sémantique.
Nous avons donc choisi d'orienter notre étude sur les grammaires d'unification et nous avons exploré les divers formalismes proposés par ce courant.

1.3 Principaux formalismes de grammaire d'unification

La génération des grammaires d'unification a permis d'intégrer les représentations sémantiques au sein des structures syntaxiques définies dans une grammaire formelle. Dans ce courant, il existe plusieurs formalismes et modèles grammaticaux pour structurer ces données syntaxiques et sémantiques. Chaque formalisme a ses propres caractéristiques, ses points forts mais aussi ses points faibles. Nous avons retenu quatre formalismes de grammaires d'unification à grande couverture : la grammaire d'arbres adjoints (TAG), la grammaire lexicale fonctionnelle (LFG), la grammaire syntagmatique généralisée (GPSG), et la grammaire syntagmatique guidée par les têtes (HPSG). Nous présentons ces grammaires en se basant sur l'ordre chronologique de leur apparition.

1.3.1 Grammaire d'arbres adjoints (Tree Adjoining Grammar : TAG)

La grammaire d'arbres adjoints (TAG : Tree Adjoining Grammar), présentée par Joshi [Joshi et al., 1975], est un formalisme syntaxique qui permet de rendre compte des liens entre les constituants de la phrase. Ce formalisme offre un système de réécriture d'arbres dont les unités sont des arbres élémentaires. Ceux-ci sont définis par l'ensemble d'arbres initiaux et d'arbres auxiliaires :

— Un arbre initial est un arbre dont les nœuds feuilles sont soit des symboles terminaux, soit des symboles non terminaux. Les symboles non terminaux sont appelés nœuds de substitution et sont marqués par le symbole (↓).

— Un arbre auxiliaire comporte un nœud feuille étiqueté par un symbole non terminal appelé "nœud pied" et il est marqué par le symbole (*). Le nœud pied et la racine de l'arbre auxiliaire sont nécessairement de même catégorie.

Les deux opérations de réécriture d'arbres autorisées par TAG sont : l'adjonction et la substitution. A l'issue de ces opérations de réécriture, on obtient un nouvel arbre appelé arbre dérivé.
L'opération de substitution, illustrée par l'exemple de la figure 1.5, permet d'insérer un

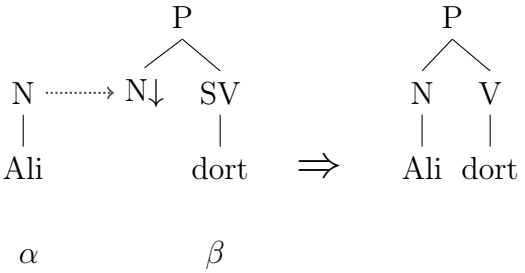


FIGURE 1.5: Substitution de l'arbre α dans l'arbre β

arbre α à la frontière d'un arbre dérivé initial contenant un nœud de substitution β. En d'autres termes, le nœud de substitution de β est remplacé par le nœud racine de α.
La substitution est autorisée uniquement si le nœud de substitution et le nœud racine, respectivement de β et de α sont étiquetés par un symbole identique.
La figure 1.6 illustre un exemple de l'opération d'adjonction. Cette opération permet d'insérer un arbre auxiliaire γ dans un arbre β sur un nœud interne X. Le nœud X situé dans β est remplacé par la racine de γ. L'adjonction est autorisée si la catégorie du nœud X est identique à la catégorie du nœud racine de γ.

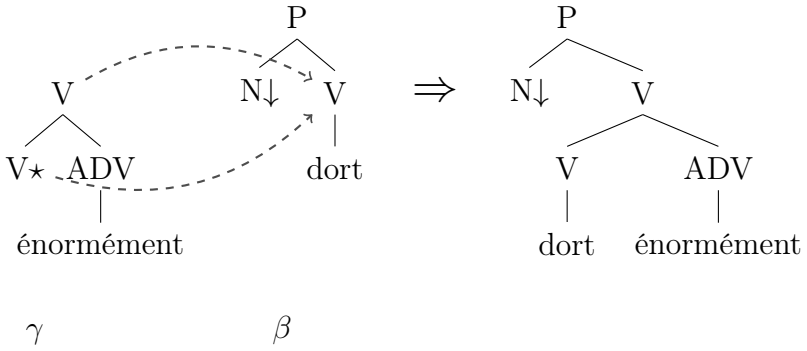


FIGURE 1.6: Adjonction de l'arbre γ au sein de l'arbre β

Shanker et Joshi ont, ensuite, intégré les structures de traits et la notion d'unification aux TAGs. D'où l'apparition de la grammaire d'arbres adjoints à base d'unification (UTAG : Unified-Based Tree Adjoining Grammar) [Vijay-Shanker and Joshi, 1991]. A chaque nœud d'un arbre élémentaire, on associe deux structures de traits nommées : partie "amont" (en anglais "top") et partie "aval" (en anglais "bottom").
La partie "amont" contient les traits indiquant les relations du nœud avec les nœuds qui le dominent, ou du même niveau. Tandis que les traits de la partie "aval", indiquent les relations du nœud avec ses nœuds fils. Par conséquence, ces structures de traits contraignent les dérivations des manières suivantes :
En cas de substitution, seuls les traits amont du nœud racine de l'arbre substitué sont unifiés avec les traits amont du nœud site de substitution, comme le montre la figure suivante.

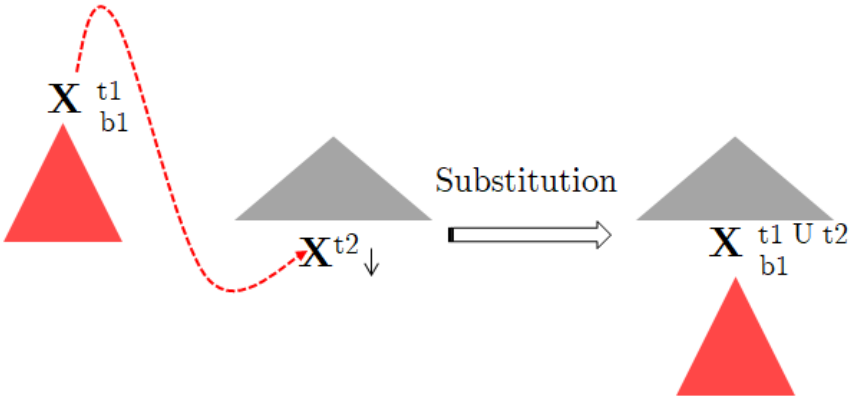


FIGURE 1.7: Unification des traits en cas de substitution [Candito, 1999]

En cas d'adjonction, les traits amont du nœud racine de l'arbre auxiliaire sont unifiés avec les traits amont du nœud site d'adjonction, et les traits aval du nœud pied de l'arbre auxiliaire sont unifiés avec les traits aval du nœud site d'adjonction. Cette opération est illustrée par la figure 1.8. En fin de dérivation, les structures de traits amont et aval de

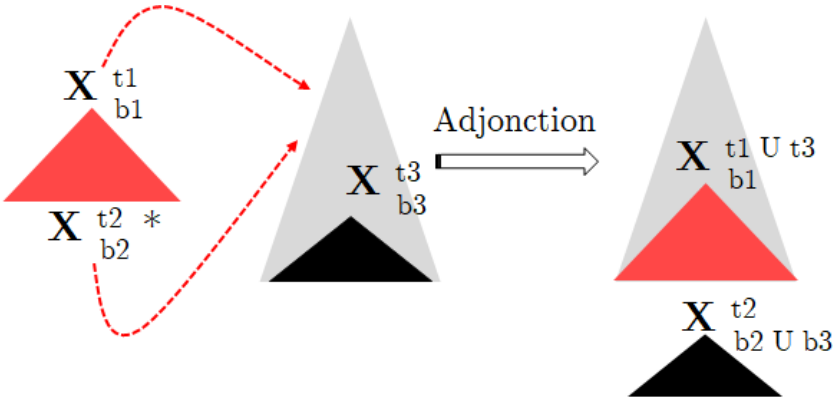


FIGURE 1.8: Unification des traits en cas d'une adjonction [Candito, 1999]

chacun des nœuds de l'arbre dérivé sont unifiées. L'unification entre deux structures de traits produit une structure résultante sauf si les deux structures portent des informations incompatibles, dans ce cas, l'unification échoue. Des principes de bonne formation des arbres élémentaires ont été rajouté [Abeillé, 2002] :

— Principe d'ancrage lexical : Tout arbre élémentaire a au moins un nœud ancre.

— Principe de cooccurrence prédicat-arguments : Chaque prédicat contient dans sa structure élémentaire au moins un nœud correspondant à son argument qu'il sous-catégorise (nœud de substitution ou nœud pied).

— Principe d'ancrage sémantique : à chaque arbre syntaxique élémentaire, un correspond sémantique non vide lui est associé.

— Principe de compositionnalité : un arbre élémentaire correspond à une et une seule unité sémantique.

Contrairement aux règles de réécriture, [8] TAG permet de représenter des arbres élémentaires de profondeur quelconque. Elles définissent un domaine de localité étendu. Ce dernier offre la possibilité de décrire certains phénomènes linguistiques telle que la représentation locale des dépendances syntaxiques.
TAG a été étendue en plusieurs extensions. Parmi ces différentes extensions nous pouvons citer TAGs lexicalisées (LTAG : Lexicalized Tree Adjoining Grammar) [Schabes and Joshi, 1990] qui permettent d'exprimer facilement les spécificités de chaque mot du lexique en lui associant un arbre représentant l'usage de ce mot dans les différentes phrases de la langue, les TAGs ensemblistes ou à composantes multiples (MCTAG : Multicomponent TAG) [Lichte, 2007] ; [Weir, 1988] pour représenter des phénomènes linguistiques tel que la permutation de complément ou encore les TAGs synchrones (STAG : Synchronous TAGs) [Shieber and Schabes, 1990] dans lesquelles des paires d'arbres élémentaires (composées d'un arbre élémentaire sémantique et un arbre élémentaire syntaxique) sont manipulées.

1.3.2 Grammaire lexicale fonctionnelle (Lexical-Functional Grammar : LFG)

La grammaire Lexicale Fonctionnelle (LFG) [Bresnan and Kaplan, 1982] a été définie par Joan BRESNAN et Ronald KAPLAN en 1982. Dans ce modèle, la phrase est décrite par deux niveaux distincts : La structure des constituants (appelée structure-c ou c-structure) et la structure fonctionnelle (appelée structure-f ou f-structure).
La structure-c reproduit la structuration de la phrase en syntagmes. Elle est représentée par un arbre syntagmatique construit par des règles indépendantes du contexte. La figure 1.9 illustre un exemple de structure-c de la phrase : "Ali lit un livre".

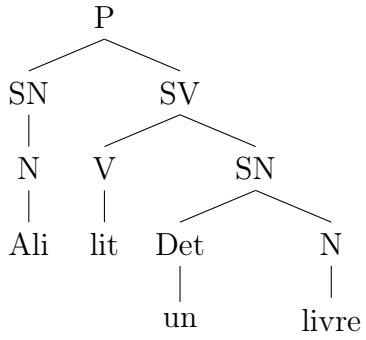


FIGURE 1.9: Exemple de structure-c

La structure-f encode les fonctions grammaticales d'une phrase (prédicat, sujet, objet, etc.) et se présente sous forme d'une structure de traits. Cette structure de trait code directement les différentes fonctions grammaticales ainsi que d'autres informations exprimant les relations entre les constituants de la phrase (exemple : les marques d'accord, certaines coréférences).

8. Une règle de réécriture est traduite par un arbre de profondeur 1

Reprenons l'exemple de la phrase précédente : "Ali lit un livre". La structure-f de cette phrase est décrite dans la figure 1.10.

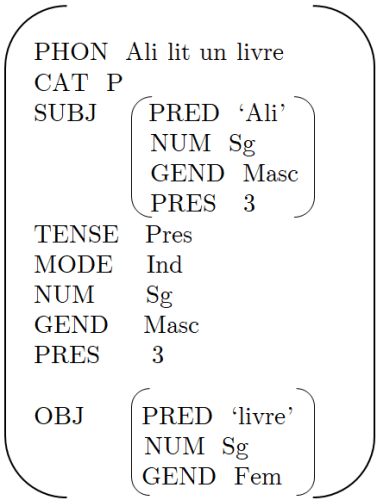


FIGURE 1.10: Exemple de structure-f

Le trait PRED représente la forme sémantique. La valeur de ce trait commence par un identifiant de l'unité sémantique véhiculée par la tête de la structure (PRED "Ali" et PRED "Livre") ou d'un cadre de sous-catégorisassions, constitué d'arguments syntaxiques séparés par des virgules (PRED : Lire <SUBJ, OBJ>). Ces arguments doivent apparaitre dans la structure-f. On dit que ces arguments (ou fonctions) sont gouvernés par le prédicat du groupe.
Chaque trait de la structure est composé d'un couple <attribut, valeur>. Ces dernières peuvent être définie par des valeurs atomiques (exemple : Sg pour l'attribut NUM, Masc pour l'attribut GEND, 3 pour l'attribut PERS, Pres pour l'attribut TENSE, etc.) ou d'une structure de traits enchâssée (exemple les valeurs des attributs SUBJ et OBJ constituées de listes de valeurs entre accolades).
La figure 1.11décrit le schéma général de LFG.

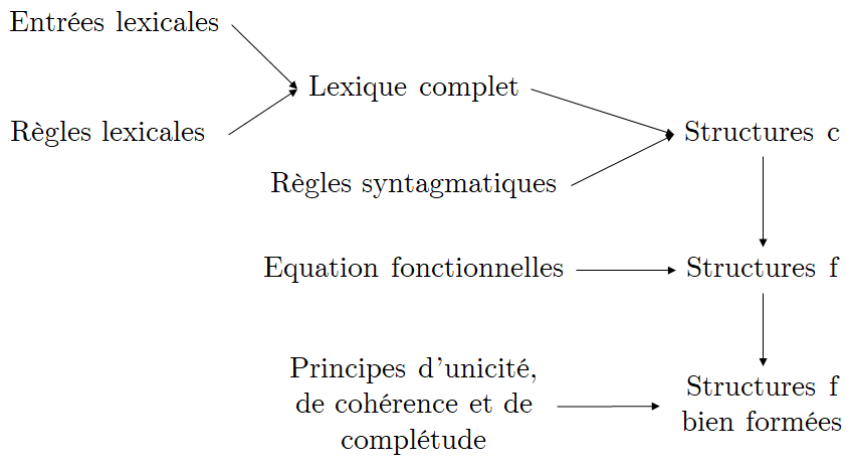


FIGURE 1.11: Schéma général d'une grammaire LFG[Abeillé, 1993]

Les règles syntagmatiques permettent de générer les constituants immédiats d'une catégorie complexe. Chaque symbole de ces règles peut être associé à zéro, une ou plusieurs équations fonctionnelles. Au sein de ces équations le symbole ↓ renvoie à la structure-f du groupe ou de la catégorie désignée (le non-terminal associé à l'équation fonctionnelle) tandis que ↑ renvoie à la structure-f du groupe immédiatement dominant dans la structure-c (qui est aussi le nœud de la partie gauche de la règle).
Ainsi l'analyse au moyen de LFG revient à construire la représentation associée à un énoncé, et à prédire si cet énoncé est grammaticalement correct ou non. Ceci revient à vérifier la bonne formation de la structure-f suivant les trois contraintes suivantes :

— Principe de complétude : La structure-f doit contenir toutes les fonctions gouvernées par leur prédicat y compris les fonctions sous-catégorisées. Par exemple la phrase "Ali donne" est incorrecte puisqu'elle ne respecte pas le principe de complétude.

— Principe de cohérence : Toutes les sous-structures doivent être localement cohérentes. En d'autres termes, les fonctions sous-catégorisables doivent être gouvernées par le prédicat local. Par exemple dans la phrase "Ali reste le livre" le principe de cohérence n'est pas respecté.

— Principe d'unicité : un attribut fonctionnel ne peut pas apparaître deux fois au même niveau dans une f-structure. Il doit avoir une valeur unique.

Le fait de dissocier la structure fonctionnelle de la structure des constituants permet de traiter de la même manière les langues à ordre fixe et les langues à ordre libre. La grammaire LFG a été appliquée à plusieurs langues telles que l'anglais, le français, l'allemand, le russe ou encore l'irlandais et aussi sur les langues indiennes, le japonais [Masuichi et al., 2003] et le coréen.

1.3.3 Grammaire syntagmatique généralisée (Generalized Phrase Phrase-Structure Grammar : GPSG)

La troisième grammaire, dans l'ordre chronologique, est la grammaire syntagmatique généralisée. Elle a été introduite par de G. Gazdar, E. Klein, G. Pullum et I. Sag en 1985 [Gerald et al., 1985] dans le but de construire une grammaire hors contexte pour les langues naturelles. Dans ce modèle, les constituants syntaxiques sont décrits à l'aide de règles syntagmatiques enrichies et plus complexes que de simples règles de réécriture. Ces règles sont de deux types DI/PL.
Les règles de Dominance Immédiate (DI) expriment la relation de dominance entre un syntagme et ses constituants immédiats. En prenant l'exemple de la règle P → SN, SV. La partie gauche (P) est appelée catégorie mère et les catégories (SN, SV) de la partie droite sont appelées les catégories filles.
Les règles DI sont divisées en deux groupes de règles : Les règles DI lexicales qui permettent d'associer à un mot du dictionnaire un ensemble de traits et les règles DI non lexicales.
Les règles de Précédence Linéaire (PL) : déterminent l'ordre des mots. Reprenons l'exemple précédent P → SN, SV. La virgule qui sépare SN et SV indique qu'il n'ya pas d'ordre de précédence entre les constituants immédiats de P. En d'autres termes la règle P → SN, SV est équivalente à P → SV, SN. Par conséquence, la définition d'une seule règle permet de représenter toutes les positions possibles d'un constituant au sein d'une structure donnée.

Afin d'imposer la précédence de SN sur SV il suffit de rajouter cette règle de précédence : SN < SV.
GPSG introduit un ensemble de métarègles qui s'applique aux règles DI lexicales. Elles sont similaires aux transformations de la grammaire transformationnelle. Par exemples la définition de métarègle(s) pour le passage en forme passive [Abeillé, 1993]. Les catégories dans GPSG sont représentées par un ensemble de traits spécifiés en utilisant la notation <attribut, valeur> (par exemple <NUM, SG>, <PERS, 3>, <CAS, DAT>). En plus de la syntaxe, GPSG associe à chaque constituant une interprétation sémantique permettant ainsi de calculer le sens compositionnel de la phrase. Cette dernière aura pour interprétation une valeur booléenne (vraie ou fausse).

1.3.4 Grammaire syntagmatique guidée par les têtes (Head-Driven Phrase Structure Grammar : HPSG)

La grammaire syntagmatique guidée par les têtes [Pollard and Sag, 1994] a été introduite par Carl Pollard et Ivan Sag en 1994. Elle se situe dans le prolongement direct des grammaires syntagmatiques généralisées GPSG (présentées plus haut), néanmoins elle intègre des innovations d'autres théories dont la grammaire LFG, la grammaire chomskyenne et les grammaires catégorielles (CG). HPSG adopte la notion de signe afin d'organiser les catégories lexicales ou syntagmatiques. En effet, chaque objet linguistique correspond à un signe. Ce dernier est défini par une structure de trait contenant l'ensemble de ses informations spécifiques (phonétiques, syntaxiques et sémantiques) et ses relations avec les autres signes de la structure. Cette organisation est appelée hiérarchie typée (ou hiérarchie de types).
La figure 1.12 illustre l'ensemble de traits généralement présents dans la description d'un signe.

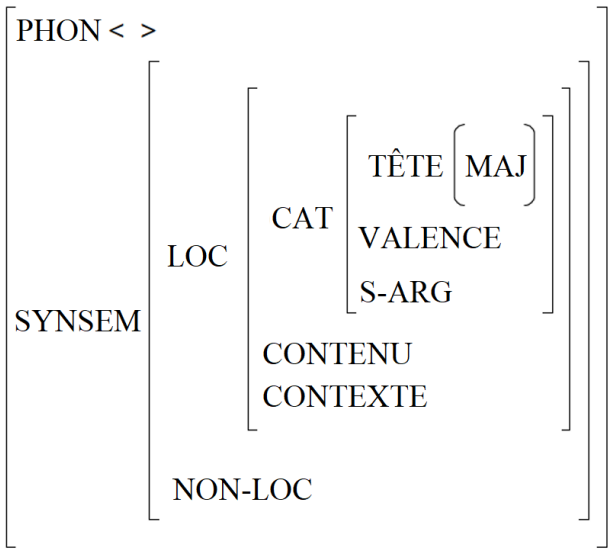


FIGURE 1.12: Ensemble des traits d'un signe linguistique[Pollard and Sag, 1994]

La description phonologique est définie via le trait PHON. La valeur de ce trait consiste

à une liste de séquences phonémiques. Quant à la syntaxe et la sémantique, elles sont présentées dans la partie "synsem" du signe. SYNSEM comporte les informations dites "locales" (trait LOC) portant sur :

— La catégorie (CAT) : elle contient au moins le trait de tête HEAD, qui inclut les informations sur la partie du discours (part of a speech/ POS) [9] du signe, et les informations de valence (SUBCAT). Ce trait de VALENCE désigne les arguments (syntaxiques) du signe. Considérons l'exemple de l'entrée lexicale suivante :

$$\begin{bmatrix} \text{PHON} & \text{<Lire>} \\ \text{HEAD} & \begin{bmatrix} \text{VFORM fin} \\ \text{Verbe} \end{bmatrix} \\ \text{SUBCAT} & \text{<SN[nom], SN[acc]>} \end{bmatrix}$$

FIGURE 1.13: Exemple de l'entrée lexicale "Lire"

La description de ce signe indique qu'il s'agit du verbe "lire". Ses informations concernent donc sa forme verbale VFORM (infinitif) et qu'il doit avoir deux arguments SN dont l'un est nominatif et l'autre est accusatif.

— Le contenu (CAT) : inclut les informations sémantiques.

— Le contexte (CTXT) : inclut les informations pragmatiques, relatives à la situation d'énonciation de l'entrée lexicale.

HPSG propose une organisation des connaissances linguistiques modularisée et non redondante. Elle repose sur l'idée de scinder la description grammaticale d'une langue en principes, règles et éléments du lexique. HPSG reprend la notion de grammaire syntagmatique avec la distinction entre règles DI/PL du modèle GPSG et élimine la notion de métarègles au profil des règles lexicales. Ces règles lexicales s'inspirent des règles présentes dans LFG et permettent de modifier les traits phonétiques, morphologiques, syntaxiques ou sémantiques des entrées lexicales.
Par ailleurs, le nombre de règles DI est réduit à un ensemble de représentations appelées : les schémas de domination immédiate. Ils correspondent aux descriptions des structures et sont au nombre de cinq : le syntagme saturé avec complément(s), le syntagme non saturé avec complément(s), le syntagme avec ajout(s) , le syntagme avec marqueur et le syntagme avec élément antéposé Ces schémas permettent de définir la bonne formation des structures et de gérer leur partage. Ils indiquent un certain type de relation entre une racine (une mère) et ses descendants (ses filles). Cependant, ces schémas ne spécifient pas l'ordre des filles ni comment les traits sont propagés à l'intérieur du syntagme. Ceci est assuré par des principes distincts représentés sous formes de structures de traits : le principe des traits de tête, le principe de sous-catégorisation, le principe sémantique, le principe du trait Spec, le principe du trait Marque, le principe d'ordre des mots et le principe des traits non locaux.

9. La valeur d'une partie du discours peut être un nom, verbe, adjectif, adverbe, préposition, etc.

1.3.5 Etude comparative

La grammaire d'unification se distingue des autres formalismes grammaticaux par une représentation plus riche des structures puisqu'elle permet d'intégrer les informations sémantiques en plus de la syntaxe. Nous consacrons cette section à une étude comparative des quatre formalismes, de ce type de grammaire, que nous avons exposés à savoir : la grammaire d'arbres adjoints (TAG), la grammaire lexicale fonctionnelle (LFG), la grammaire syntagmatique généralisée (GPSG et la grammaire syntagmatique guidée par les têtes (HPSG).

Critères / **Formalismes**	**Représentation des connaissances**	**Dépendance contextuelle**	**Représentation de la sémantique**	**Pouvoir de croisement**	**Complexité des traitements**
TAG	Arbres élémentaires + Les structures de traits (pour UTAG)	Lègère dépendance du contexte	Oui	Oui	Polynomial $O(n^6)$
LFG	Structure de constituants + structure fonctionnelle	Forte dépendance	Oui	Non	NP-complet exponentiel
GPSG	Règles de dominance immédiate (métarègle) + règles d'ordre linéaire	Hors contexte	Oui	Non	NP-complet
HPSG	Règles lexicales + les schémas de domination immédiate	Hors contexte	Oui	Non	NP-complet exponentiel

TABLE 1.1: Comparatif des différents formalismes des grammaires d'unification

Les critères de comparaison que nous avons choisis sont les suivants [Ben Fraj, 2010] :

— La représentation des connaissances : c'est le procédé utilisé pour organiser les informations au sein de la grammaire ;

— La dépendance contextuelle : c'est la classe des grammaires en considérant la classification des langages de Chomsky ;

— La représentation de l'information sémantique : c'est la capacité de la grammaire à représenter les descriptions sémantiques pouvant être partagées entre les différentes composantes syntaxiques de la structure représentée ;

— Le pouvoir de croisement : c'est la capacité de représenter les structures croisées telles que les subordonnées ;

— La complexité du traitement : c'est la quantité de ressources, en temps et en espace, nécessaire pour vérifier l'appartenance d'une phrase au langage engendré par une grammaire. Elle est exprimée en fonction de n, le nombre des terminaux de la phrase.

Ces quatre formalismes appartiennent à des catégories grammaticales différentes. La grammaire GPSG et la grammaire HPSG appartiennent à la classe des grammaires hors contexte. Ce type de grammaire n'offre pas une bonne modélisation de la syntaxe pour les traitements linguistiques [Joshi, 1987]. Dans HPSG, la distinction entre les différents niveaux (répartition en trois ou quatre niveaux) de représentation dans ces structures est peu claire. De plus, la structure en constituants obtenue est induite par le processus d'analyse plutôt que par la grammaire elle-même [Pollard and Sag, 1994]. En revanche, la grammaire LFG est une grammaire contextuelle et la grammaire TAG est légèrement sensible du contexte (mildly context sensitive). Ces deux formalismes offrent des représentations de structure plus simples que les structures des grammaires syntagmatiques. LFG distingue deux représentations syntaxiques : les structures configurationnelles et fonctionnelles mais privilégie le niveau lexical-fonctionnel. Les unités de la syntaxe traitées sont les mots et non les morphèmes. Tandis que, dans TAG, les règles de réécriture sont remplacées par des arbres élémentaires. Ensuite, les arbres dérivés sont générés à la suite de l'application des opérations d'adjonction ou substitution. La force de représentation de TAG réside dans l'ensemble des restrictions sur les combinaisons de ses structures. Elle permet ainsi une meilleure gestion et manipulation de leur construction. Cette caractéristique la rend plus puissante que les grammaires hors contexte. L'un des avantages de TAG, et qu'elle est légèrement contextuelle. Cette classe de grammaire, permet d'engendrer tous les langages hors-contexte ainsi que certains langages contextuels. De ce fait, ce type de grammaire permet l'analyse des dépendances croisées, phénomène qui ne peut être représenté avec une grammaire hors contexte [Lecomte, 2004].
Enfin, nous remarquons que concernant la complexité des grammaires, les grammaires syntagmatiques HPSG, GPSG et la grammaire LFG présentent une complexité élevée qui n'est pas triviale à gérer dans des analyseurs syntaxiques. Une grammaire hors-contexte s'analyse en un temps polynomial borné par O (n^3). Puisque GPSG a la même capacité générative qu'une grammaire hors-contexte on peut penser qu'elle s'analyse aussi en un temps polynomial. Cependant le problème de l'analyse syntaxique dans le formalisme DI/PL est NP-complet et son analyse se fait en un temps exponentiel [Ristad, 1987]. En revanche, TAG possède suffisamment de contraintes lui permettant d'éviter les problèmes d'indécidabilité et de complexité du traitement. Ainsi, elle permet une analyse en un temps polynomial proportionnel à la longueur de la phrase en entrée.

1.4 Principaux formalismes de représentation pour la sémantique formelle

Du point de vue de la sémantique formelle ou encore compositionnelle, les informations sémantiques peuvent être représentées de différentes manières. Nous avons focalisé notre étude sur les principaux formalismes de représentation et de calcul sémantique. Dans cette section nous présentons un panorama de ces formalismes qui reposent sur le principe de compositionnalité et assurent l'interfaçage entre la syntaxe et la sémantique, à savoir le langage du calcul des prédicats, le lambda-calcul typé (λ-calcul typé), la sémantique à trous et les cadres sémantiques.

1.4.1 Langage du calcul des prédicats

Les travaux de Richard Montague [Montague, 1974b], [Dowty et al., 1981] ont introduit le premier système de calcul sémantique automatisable pour la langue naturelle. Montague propose de traiter le langage naturel comme un système formel interprété : *"Il n'y a selon moi aucune différence théorique importante entre les langues naturelles et les langages artificiels des logiciens ; en effet, je considère que l'on peut comprendre la syntaxe et la sémantique de ces deux types de langage au sein d'une même théorie naturelle et mathématiquement précise"* [Montague, 1970]. Ce système est basé sur le calcul des prédicats. Les éléments de ce langage sont les suivants :

— Les prédicats : définis sous forme de symboles qui désignent des relations entre des individus prédéfinis. Par exemple, un symbole prédicat peut traduire un verbe (dort, aime, donne).

— Les arguments : sont les individus auxquels est appliqué le prédicat. Par exemple, un prédicat qui traduit le verbe "aimer", a besoin de deux arguments pour traduire correctement des constructions transitives. En d'autres termes, les arguments du prédicat sont équivalents aux sujet et l'objet de la construction syntaxique du verbe transitif "aimer".

— Les opérateurs logiques : $\neg$ (négation), $\wedge$ (et), $\vee$ (ou), $\rightarrow$ (implication matérielle).

— Les quantificateurs : $\exists$ (existe), $\forall$ (pour tout)

La relation entre un prédicat et ses arguments est caractérisée par une propriété appelée l'arité [10] du prédicat. Cette dernière désigne le nombre d'arguments qu'un prédicat prend. Par exemple, considérons le verbe transitif "aimer" et l'ensemble de constantes suivant = {Ali, Fatima}. Le prédicat définit pour ce verbe est AIMER(x,y), où x et y sont des variables qui désignent respectivement le sujet et objet. Grâce à ces éléments, nous pouvons décrire les phrases suivantes :
[[Ali aime Fatima]] [11] = AIMER(Ali,Fatima)
[[Fatima n'aime pas Ali]] = $\neg$ AIMER(Fatima,Ali)
Montague s'est basé sur la logique des prédicats pour mettre en place un système de calcul sémantique compositionnel pour l'anglais. Ce mécanisme se résume comme suit [Montague, 1974a] : Tout d'abord, il est primordial de définir l'ensemble de règles pour la description de la structure syntaxique des phrases couvertes. Ensuite, à chaque mot du lexique une représentation sémantique est associée. Pour se faire, Montague définit deux types sémantiques de base : e (type des individus) et t (type des valeurs de vérités). Enfin, les règles syntaxiques vont s'accoupler avec les règles sémantiques afin de guider le calcul sémantique.
Afin d'apporter plus d'explication, considérons l'application de ce principe sur une structure syntaxique arborescente. Au début de l'analyse sémantique, seuls les nœuds terminaux possèdent un type. Ces nœuds forment l'ensemble des éléments lexicaux assignés chacun à un type (e et/ou t). Au fur et à mesure de l'analyse de la phrase, les types s'attribuent aux nœuds intermédiaires suivant les règles de typage. [12] Les travaux de Montague

10. Si un prédicat attend n arguments, on dit que son arité est n ou bien que le prédicat est n-aire (ou encore à n places).

11. La notation [[ ]] sert à lier une phrase à sa formule logique.

12. Les types sont composés à condition toutefois que les types soient compatibles.

ont influencé beaucoup d'approches pour le traitement automatique de la sémantique du langage dont celle basée sur le lambda-calcul typé (λ-calcul typé).

1.4.2 λ-calcul typé

Le λ-calcul [13] typé peut être considéré comme une extension fonctionnelle du calcul des prédicats par l'introduction du lambda opérateur. La représentation sémantique de la phrase est calculée à partir des λ-termes typés des mots la composant. Par exemple, la phrase "Ali aime Fatima" est composée de ces trois éléments :

— Ali annoté par le type : <e>

— λx.λy.aime(x,y) annoté par le type : <e,<e,t»

— Fatima annoté par le type : <e>

Où la variable x réfère à Ali et y à Fatima. Les dépendances obtenues par la structure arborescente, résultat de l'analyse syntaxique (voir figure 1.14), sont converties en une suite de termes λ-calcul typé. Les prédicats, qui forment les nœuds de la structure, s'appliquent à leurs arguments suivant l'ordonnancement défini.

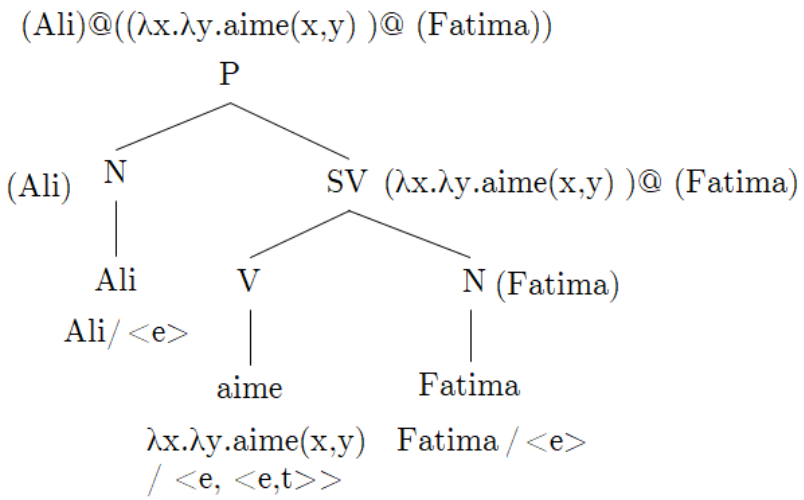


FIGURE 1.14: Exemple de calcul pour la phrase "Ali aime Fatima"

L'analyse sémantique consiste ensuite au remplacement des lexèmes par les termes qui leur correspondent dans le lexique et à effectuer des opérations de bêta réduction (β-réduction). Ce procédé, désigné par le symbole @, permet d'instancier un argument de la fonction. La formule logique finale obtenue représente le sens calculé. La formule sémantique de la représentation syntaxique illustrée par la figure 1.14. est calculée comme suit :
(Ali)@((λx.λy.aime(x,y))@ (Fatima))
((λx.(λy.(aime (x,y))))@(Ali))@(Fatima)
(λy.(aime (Ali, y)))@(Fatima)
aime(Ali, Fatima)

13. Le λ-calcul est une façon abstraite de définir une fonction et ses arguments.

1.4.3 Sémantique à trous

Le formalisme des sémantiques à trous (Hole Semantics) [Bos, 1995] permet la réduction des ambiguïtés de portée [14] via un procédé de sous-spécification. L'idée est d'associer à une phrase une unique représentation sémantique sous-spécifiée, qui sera résolue pour produire toutes ses interprétations possibles pour la phrase. Les objets utilisés dans cette représentation sont les suivants :

— des constantes d'étiquette notées li ;

— des variables d'étiquette, appelées trous (holes) et notées hj ;

— des contraintes de portée, représentées par l'opérateur $\leq$.

Les solutions d'une formule sous-spécifiée représentées au moyen de ce formalisme, sont réalisées par l'ensemble des injections entre constantes et variables d'étiquettes respectant l'ensemble des contraintes imposé par cette formule. Considérons l'exemple de la phrase présentant une ambiguïté de portée "Tout homme aime une femme". La représentation de cette phrase au moyen de la sémantique à trous est la suivante :

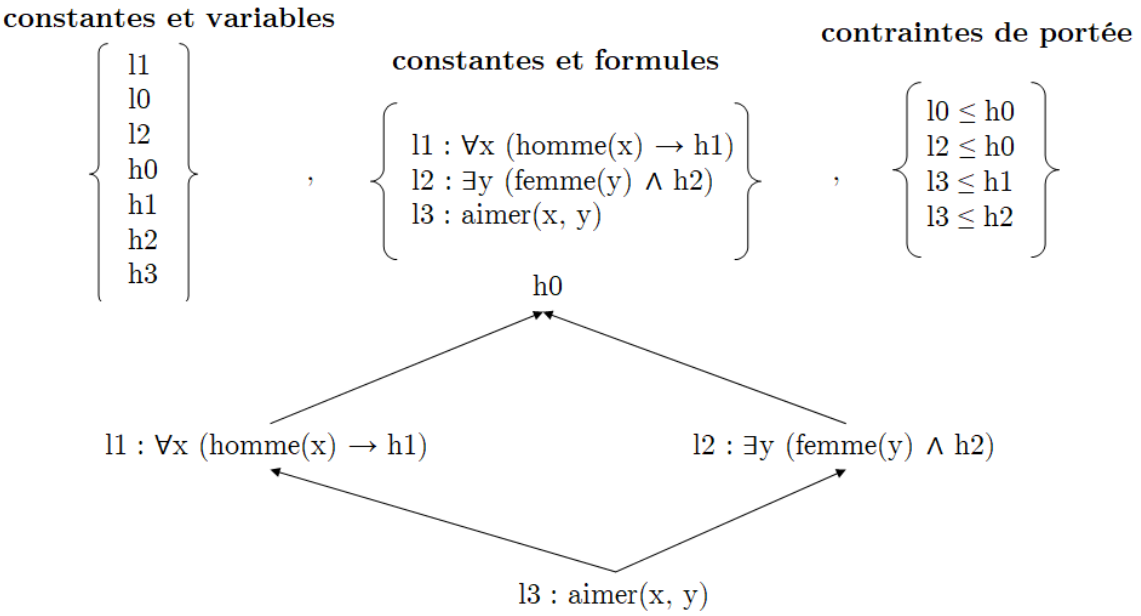


FIGURE 1.15: Représentation au moyen de la sémantique à trous de "Tout homme aime une femme"

Le calcul de la sémantique revient à interpréter le(s) branchement(s) obtenu(s) entre les étiquettes et leurs contraintes. Les formules décrites dans la figure 1.15 produisent deux branchements [15] possibles :

— P1 : l1=h0,l2=h1,l3=h2 qui permet de construire la construction sémantique suivante : ∀ x.homme(x) → (∃ y.femme(y) ∧ aimer(x, y)).

14. Une ambigüité de portée correspond à une situation où deux quantificateurs (ou expressions au comportement similaire) peuvent être interprétés comme ayant une portée supérieure l'un par rapport à l'autre.

15. P pour plugging en anglais.

— P2 : l2=h0,l3=h1,l1=h2 qui se traduit par la représentation suivante : ∃ y.femme(y) ∧ (∀ x.homme(x) → aimer(x, y)).

Nous notons que ce formalisme permet de traiter les prédicats dont les arguments sont soit des constantes soit des variables et donc, des termes non-récursifs. Ainsi, la sémantique à trous fait partie de ce que l'on appelle les sémantiques plates. Ce format de représentation sémantique a été appliqué dans différents travaux tels que les travaux de [Gardent and Kallmeyer, 2003] pour l'anglais. Dans cette approche, la sémantique plate utilise la logique fondée dans [Bos, 1995], augmentée de variables d'unification [Gardent, 2006]. Le principe consiste à associer des formules logiques du premier ordre aux arbres élémentaires. Ces formules vont être composées lors de l'analyse de phrases pour former sa représentation sémantique.

1.4.4 Cadres sémantiques

Fillmore [Fillmore, 1982] définit les cadres (frames) comme des structures cognitives représentant tout système relationnel de concepts au sein duquel la compréhension d'un concept fait appel à la compréhension du système complet. Les cadres sémantiques sont basés sur une théorie antérieure proposée par Fillmore [Fillmore, 1967] appelée grammaire de cas (case grammar or case-based grammar). Cette théorie définit un ensemble de cas universels qui permet de mettre en avant la relation entre un verbe et ses composants nominaux. Un cadre peut être défini par une structure de données représentant un concept en associant à son nom un ensemble d'éléments décrivant ses rôles situationnels (attributs) ou relationnels (rôles sémantiques). En effet, divers rôles sémantiques peuvent exister dans le langage (exemple : acteur, agent, source, destination, etc.). Considérons le verbe transitif "aimer". Ce dernier doit pouvoir assigner des rôles différents aux syntagmes nominaux remplissant les fonctions de sujet et d'objet. Dans cette situation, le sujet du verbe représente l'individu qui fait l'expérience décrite par le verbe "aimer". Ce rôle est appelé experiencer. Tandis que l'objet direct est celui qui "reçoit" et qui est appelé le patient. De cette manière pour la phrase "Ali aime Fatima", on a la représentation de cadre sémantique suivante :

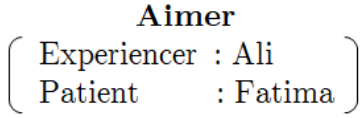


FIGURE 1.16: Cadre sémantique de la phrase "Ali aime Fatima"

De même, la phrase "Fatima aime Ali" est représenté par le cadre sémantique suivant :

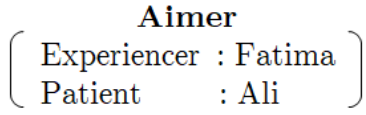


FIGURE 1.17: Cadre sémantique de la phrase "Fatima aime Ali"

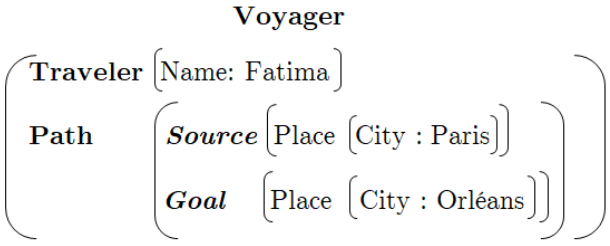


FIGURE 1.18: Cadre sémantique de la phrase "Fatima voyage de Paris vers Orléans"

Les rôles sémantiques permettent donc d'établir un lien entre la grammaire et la signification. Contrairement aux représentations plates, l'usage des cadres sémantiques offre une représentation hiérarchiquement structurée. La figure 1.18 illustre un exemple de représentation sémantique de la phrase "Fatima voyage de Paris vers Orléans". En effet, le verbe ditransitif "voyager" possède trois arguments auxquels on devra attribuer leurs rôles. Ces trois rôles sont : Traveler (le voyageur), Source (le point de départ) et Goal (but ou encore le point d'arrivée). Les deux derniers rôles constituent le chemin du voyage. La correspondance entre les arguments syntaxiques de la phrase et les rôles sémantiques du cadre est réalisée comme suit : La voyageuse (Traveler) s'appelle Fatima. La direction est décrite grâce au chemin (Path). Ce dernier est aussi représenté au moyen de cadres sémantiques spécifiant que le lieu source est la ville "Paris" vers le la destination finale (Goal) la ville "Orléans".

1.4.5 Discussion comparative des formalismes sémantiques

Nous avons présenté dans cette section, quatre formalismes de représentation et de calcul sémantique. Toutes ces méthodes peuvent intervenir dans l'interfaçage entre la syntaxe et la sémantique d'une langue donnée.
Le λ-calcul est considéré comme étant un formalisme standard pour les recherches concernant la sémantique computationnelle. Néanmoins, tout comme la logique des prédicats, ce procédé s'avère être insuffisant pour traiter des représentations sémantiques plus complexes, au sein des grammaires de grande taille, telles que les ambigüités de portée, la décomposition des structures sémantiques ou encore les relations entre ces structures sémantiques partiels. La sémantique à trous a permis d'apporter un ensemble de solution à ces limites puisqu'elle comble le problème de gestion de l'ambiguïté de portée. Cependant elle est assez lourde à manipuler.
Les cadres sémantiques offrent une représentation sémantique riche et structurée. Seulement, il est difficile d'intégrer des opérateurs logiques tels que les quantificateurs dans les cadres. Ils sont plus adéquats pour capturer le sens lexical structuré et les différences de sens subtiles. En effet, ce formalisme assure la diversité de possibilités combinatoires des éléments des cadres. Il traite des scénarios plus complexes tels que les interrelations entre des événements distincts mais qui se produisent dans le même domaine. Enfin, grâce à ce format de représentation, il est aussi possible de relier les entrées lexicales à des formes de représentations conceptuelles. En d'autres termes, fournir une description du lexique selon les principes de cadres conceptuels.

1.5 Conclusion

Dans ce chapitre, nous avons entamé l'étude de l'état de l'art sur le processus de traitement d'une langue naturelle dans le domaine de TALN. Nous nous sommes concentrés sur les deux niveaux d'analyse syntaxique et sémantique. Ceci nous a conduit à introduire la notion de compositionnalité sémantique assuré par l'interface syntaxe-sémantique. Ce procédé repose sur la syntaxe pour guider le processus d'analyse sémantique. Nous nous sommes donc focalisés sur l'étude des formalismes grammaticaux pour la représentation des structures syntaxiques. Plus particulièrement, l'étude s'est orientée vers un ensemble de formalismes des grammaires d'unification. Parmi les différents courants de grammaires formelles, les grammaires d'unification permettent d'intégrer la sémantique au moyen des variables d'unification.
A l'issue d'une étude comparative de ces formalismes, nous avons relevé un ensemble d'avantages des grammaires TAG qui se distinguent par leur riche pouvoir de représentation (les structures simples, complexes, combinatoires, partagées, etc...) notamment leur fort pouvoir génératif, qui englobe les dépendances à longue distance et certaines dépendances croisées ainsi qu'une factorisation des composantes grammaticales récursives. De plus, contrairement aux autres grammaires précédemment présentées, TAG reste analysable en un temps polynomial O (n^6). Cette grammaire permet aussi l'interfaçage entre le niveau syntaxique et le niveau sémantique. Nous avons présenté une panoplie des méthodes de représentation de la sémantique pouvant intervenir dans la construction d'une telle interface. Comparés aux autres formalismes, les cadres sémantiques assurent une représentation plus riche et plus structurée de la sémantique.
Toutes ces caractéristiques nous ont motivés à utiliser le formalisme grammatical TAG et les cadres sémantiques dans notre travail de thèse qui vise à élaborer une grammaire formelle pour décrire la syntaxe et la sémantique de la langue arabe.
Actuellement, les outils et ressources numériques utiles pour le traitement de l'arabe sont relativement rares. La langue arabe, malgré son importance, est encore considérée comme une langue peu dotée. Ceci peut s'expliquer par les spécificités de cette langue qui rendent son traitement plus complexe et aussi par le démarrage assez tardif des travaux de recherches la concernant. Dans le chapitre suivant, nous nous intéressons à la langue arabe ainsi qu'à un ensemble de ses ressources numériques.

Chapitre 2

Langue arabe : spécificités, ressources et grammaire d'arbres adjoints

2.1 Introduction

L'arabe, qui est une langue sémitique, est la 4ième langue la plus parlée au monde [1] avec un nombre de locuteurs natifs estimé au moins à 295 millions de personnes. Elle se présente sous deux formes principales : l'arabe littéral et l'arabe dialectal.
L'arabe littéral, aussi appelé arabe classique, arabe standard moderne (ASM) ou arabe éloquent, représente la langue diffusée et enseignée dans tous les pays arabes. Elle est associée à la culture littéraire (la religion et l'écrit), aux sciences, aux médias ainsi qu'aux fonctions administratives. Toutefois, l'arabe classique d'aujourd'hui ne correspond pas exactement à la langue dans laquelle fut écrit le Coran il y a plusieurs siècles. L'arabe classique moderne utilisé de nos jours a subi certaines réformes qui ont un peu modifié et simplifié la syntaxe originale de l'arabe. Dans les pays arabes, l'arabe standard moderne est enseigné dès le primaire et constitue la langue officielle. Cependant, la langue parlée par les arabophones demeure l'arabe dialectal. Les dialectes sont des versions encore plus simplifiées de l'arabe. Il existe une grande variété d'arabe dialectal (Égyptien, Marocain, Saoudien, Tunisien, etc.). En effet, ces simplifications et variations sont opérées au niveau de la grammaire, la conjugaison et la déclinaison. Néanmoins, ils ont beaucoup de vocabulaire en commun avec l'arabe courant. De ce fait, il est plus utile et facile d'apprendre l'arabe courant avant d'apprendre un ou plusieurs dialectes.
Nous avons consacré ce chapitre à l'étude de la langue arabe dans sa version standard moderne. Dans la première partie, nous allons décrire les spécificités de cette langue. Ensuite, nous présenterons un ensemble des ressources numériques utiles pour son traitement. Enfin, la dernière partie sera consacrée aux travaux antérieurs réalisés pour la construction d'une TAG pour l'arabe. Nous nous nous intéresserons, plus particulièrement aux insuffisances de chacune de ces approches.

1. Selon le site ethnologue : www.ethnologue.com/statistics/size

2.2 Spécificité de la langue arabe

La langue arabe présente beaucoup de spécificités à différents niveaux : phonologique, morphologique, syntaxique et aussi sémantique, la rendant relativement plus difficile à traiter que les langues indo-européennes. Nous nous focalisons sur les aspects morphosyntaxique sujet de notre étude.

2.2.1 Propriétés morphologiques

Un trait distinctif de la morphologie arabe est que la formation du mot se fait via un processus de croisement et concaténation. Le vocabulaire arabe est construit en croisant des unités abstraites minimales composées exclusivement de consonnes (généralement trois consonnes), appelées racines, avec un nombre de modèles appelés schèmes. À cela s'ajoute un ensemble restreint d'affixes. Prenons l'exemple de la racine trilitère ك - ت - ب (k-t-b / écrire). Les mots suivants sont obtenus en fusionnant cette racine avec certains schèmes :

Mot	Schème	Phonétique	Traduction	Type de dérivé
كتب	C1a-C2a-C3a	Ka-Ta-Ba	il a écrit	Verbal
يكتب	yaC1-C2u-C3u	yaK-Tu-Bu	il écrit	Verbal
كاتب	C1ā-C2i-C3	Kā-Ti-B	écrivain	Nominal
مكتوب	maC1-C2ū-C3	maK-Tu-B	écrit	Nominal
مكتبة	maC1-C2a-C3ah	maK-Ta-Bah	bibliothèque	Nominal

Table 2.1: Exemple de mots dérivés de la racine k-t-b (écrire)

Dans le schème, la lettre C, en majuscule, représente la consonne. Cette dernière est décorée par un indice qui précise sa place dans la racine. Quant aux autres lettres en minuscules, elles représentent les éléments (les consonnes et/ou voyelles) du schème qui sont ajoutés à la racine.

— Les préfixes attachés au début du mot. Par exemple la lettre "ن" qui indique la première personne du pluriel dans un verbe conjugué tel que "نكتب" (nous écrivons).

— Les suffixes attachés à la fin du mot. Par exemple les deux lettres "ون" qui indique que le verbe est conjugué à la 3ème personne du pluriel du masculin tel que "يكتبون" (ils écrivent).

— Les circonfixes qui entourent le mot. Par exemple la lettre "ت" au début et les deux lettres "ين" ajoutées à la fin d'un verbe indique qu'il est conjugué à la deuxième personne du singulier du féminin. Tel que "تكتبين" (tu écris)

2.2.1.1 Formes agglutinées

Les mots en arabe peuvent être agglutinés. Le phénomène d'agglutination des mots consiste à joindre des enclinomènes (proclitiques et/ou enclitiques) aux formes simples ce qui donne lieu à des formes plus complexes dites agglutinées. Un mot en arabe peut correspondre à toute une phrase. Par exemple "سيكتبه" (il va l'écrire) est une phrase composée d'éléments proclitiques et enclitiques qui s'ajoutent à un mot minimal. En Arabe, les proclitiques dépendent de plusieurs critères pour s'attacher au début d'un mot ou d'un syntagme. Lorsqu'ils s'accolent aux verbes, ils dépendent de l'aspect verbal. A titre d'exemple nous pouvons citer la particule du futur "س" qui exprime le futur lorsqu'elle s'attache au verbe. Dans le cas du verbe "كتب" conjugué à la 3ième personne du singulier "يكتب" (il écrit) devient "سيكتب" (il va écrire). Lorsque les proclitiques s'accolent aux noms, ils dépendent de leur mode et de leur cas. Parmi les proclitiques des noms et des adjectifs nous pouvons citer l'article défini "ال" (le/la) (exemples : "الكتاب" (le livre), "الفتاة" (la fille)) et les prépositions "ك" (comme) (exemple : "كالملاك" (comme un ange)), "ب"(avec) (exemple : بسعادة (avec joie)) et "ل" (pour) (exemple : له (pour lui)).
Les particules de conjonction se placent entre معطوف عليه (le coordonné à lui) et le معطوف (le coordonné). Il existe deux types de ces particules :

- Les particules qui permettent de combiner le coordonné avec l'élément principal : "و" (et), "ف" (puis), "ثم" (donc), "حتى" (jusqu'à), "مثل" (comme) : Exemple : "أكل علي و محمد الطعام" (Ali et Mohammed ont mangé le repas). L'élément principal "علي" (Ali) et le coordonné "محمد" (Mohammed) ont partagé le même repas.

- Les particules qui permettent de choisir entre le coordonné et le coordonné à lui : "أم" (ou), "أو" (ou), "لا" (ni), "بل" (mais), "ولكن" (cependant). Exemple : "هل تفضل القطط أوالكلاب" (Est-ce que tu préfères les chats ou les chiens). Dans cet exemple un choix s'impose entre deux choses : "القطط" (les chats) (coordonnée à lui) et "الكلاب" (les chiens) (le coordonné).

Les enclitique apparaissent à la fin du mot ou d'un syntagme tels que les pronoms compléments d'objets (ه ، ك ، هما, etc.). Contrairement aux proclitiques qui ont la faculté de se combiner entre eux, on ne peut accoler qu'un seul enclitique à un mot.
Les enclitiques des verbes varient selon le mode du verbe. Ils s'accolent aux verbes transitifs pour exprimer le complément d'objet direct. Par exemple le verbe "أكلها" (l'a mangé) dont le suffixe "ها" (ha) représente le complément d'objet direct. Les enclitiques des noms expriment les pronoms possessifs. Par exemple, le pronom possessif du verbe كتب (écrire) est كتابي (mon livre). Analysons l'exemple précédent du mot "سيكتبه" (il va l'écrire). Nous remarquons qu'il est composé d'une particule du futur "س" (sa), le verbe "كتب" (écrire) conjugué à la 3ième personne du singulier du masculin "يكتب"(écrit), et un complément

d'objet "ه" (hou).

2.2.1.2 Voyelles

La voyellation consiste à placer des signes au-dessus ou au-dessous du caractère arabe pour illustrer la prononciation correcte de la lettre. Par conséquent, les voyelles permettent de soulever l'ambiguïté entre les homographes [2] et d'éviter la confusion du sens du mot lorsqu'il existe plusieurs façons de prononcer ce dernier. La langue arabe dispose de deux types de voyelles : les voyelles courtes et les voyelles longues. Elles sont représentées par des signes répartis au-dessus ou au-dessous des consonnes.
Les voyelles courtes sont au nombre de quatre :

— la fatha (الْفَتْحَة) : représentée par le trait "ـَ" sur la lettre et se prononce "a". Exemple : دَ = Da

— la kasra (الكَسْرَة) : représentée par le trait "ـِ" sous la lettre et se prononce "i". Exemple : دِ = Di

— la dhamma (الضَمةْ) : représentée par le symbole "ـُ" sur la lettre et se prononce "ou". Exemple : دُ = Dou

— le soukoune (السُكون) : il est représenté par le symbole "ـْ" sur la lettre et correspond à l'absence de toute voyelle. Exemple : دْ= aD.

Tandis que les voyelles longues sont au nombre de trois :

— le alif (ا) : utilisé pour prolonger la fatha. Exemple : دَا = Daa

— le ya (ي) : utilisé pour prolonger la kasra. Exemple : دِي = Dii

— le waw (و) : utilisé pour prolonger la damma. Exemple : دُو = Douu

Il existe aussi trois signes de nounation (تنوين/ tanwin) "ـً", "ـٍ" et "ـٌ" qui se prononcent respectivement "an","in", "un". Ils correspondent à l'ajout d'un "n" à une voyelle finale pour les noms indéterminés.

2.2.2 Propriétés syntaxiques

En arabe, il existe deux types de phrases : La phrase verbale et la phrase nominale ainsi que plusieurs types de syntagmes. Nous avons effectué une étude de ces structures en se référant à des livres de grammaire. [3]

2. Désigne les mots qui s'écrivent de la même manière mais peuvent ne pas être prononcés de la même façon.

3. Livres scolaires niveaux 7ième, 8ième et 9ième année de l'enseignement de base en Tunisie ainsi que le livre : "La grammaire Arabe pour tous" de Djamel Kouloughli

2.2.2.1 Syntagme

Le syntagme (مركب) représente un groupe de mots qui peuvent être combinés pour donner naissance à des propositions simples ou complexes.

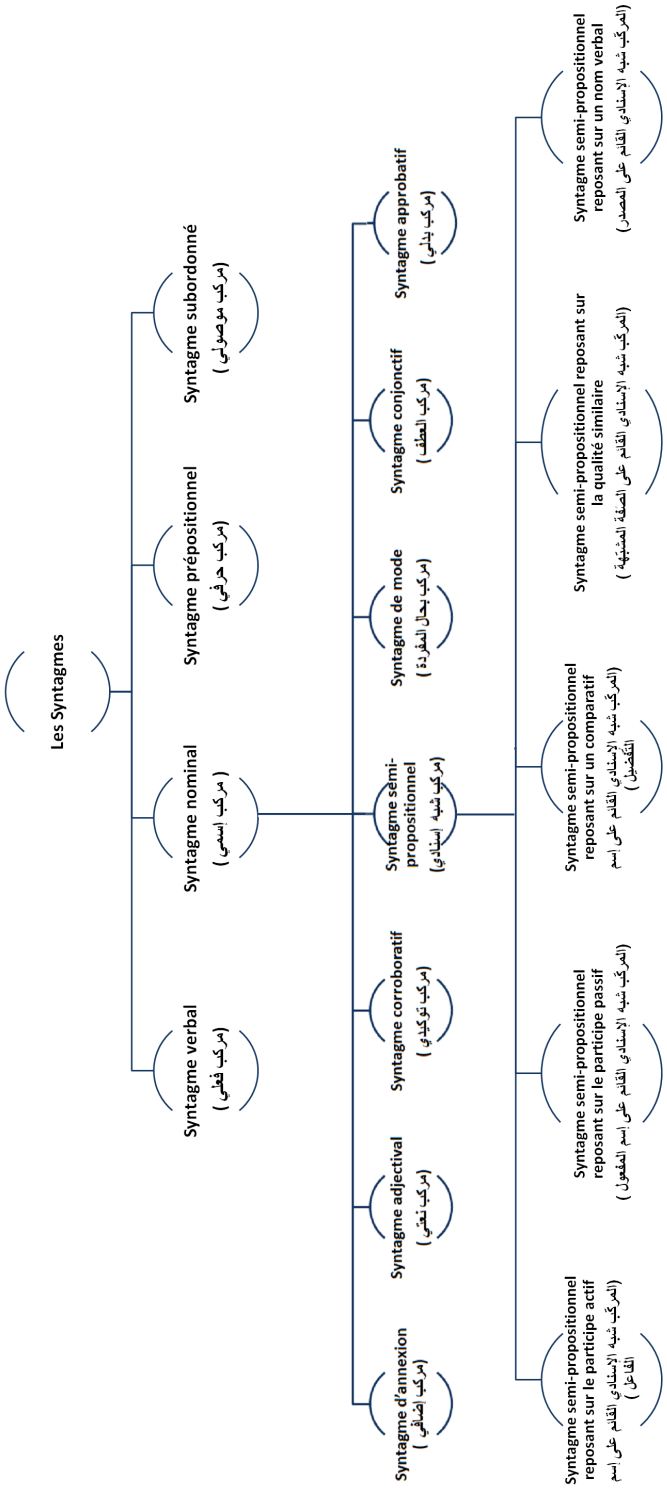


FIGURE 2.1: Hiérarchie des syntagmes en langue arabe

Ces dernières peuvent générer des phrases. La figure 2.1 illustre les différents types de syntagme en arabe.

— Le syntagme verbal (مركب فعلي) : commence obligatoirement par un verbe. Le verbe peut être est précédé par une particule : قد , سوف, ما,السين, لما, لن, لم, لا, etc.
Exemple : "لا تستصغر عدوك" (Ne sous-estime pas ton ennemi)

— Le syntagme nominal (مركب اسمي) : contient un thème qui peut être aussi un syntagme nominal et un propos qui peut être un syntagme verbal.
Le syntagme nominal possède plusieurs catégories :

— le syntagme d'annexion (مركب إضافي) : "كتاب القراءة" (livre de lecture).

— le syntagme adjectival (مركب نعتي) : "قصة ممتعة" (histoire amusante).

— le syntagme corroboratif (مركب توكيدي) : "المدعوون كلهم" (les invités tous).

— le syntagme approbatif (مركب بدلي) : "أحمد إبن الجيران" (Ahmed le fils des voisins).

— le syntagme de mode (مركب بحال المفردة) : "ماشيا" (le garçon en marchant).

— le syntagme conjonctif (مركب العطف) : "أحمد وصديقه" (Ahmed et son ami).

— le syntagme quasi-propositionnel (مركب شبه إسنادي) : "راكبا دراجته" (montant son vélo).
Il existe cinq types de syntagme quasi-propositionnel :

— Le syntagme quasi-propositionnel reposant sur le participe actif (مركب شبه إسنادي قائم على إسم الفاعل) : Le participe actif, en arabe, exprime l'acteur de l'action du verbe ou son comportement comme le verbe "ساق" (conduire) qui devient un participe actif "سائق" (conducteur).
Exemple : "سائِق الحافِلةِ" (conducteur du bus).

— Le syntagme quasi-propositionnel reposant sur le participe passif (مركب شبه إسنادي قائم على إسم المفعول) : Le participe passif, en arabe, indique le caractère de celui qui a subi l'action du verbe comme "مكتوبٌ" (est écrit) qui devient un participe passif de "كَتَبَ" (écrire).
Exemple : "مكتوبٌ في الكتابِ" (est écrit dans le livre).

— Le syntagme quasi-propositionnel reposant sur un comparatif (مركب شبه إسنادي قائم على إسم التّفضيل) : Le comparatif indique la qualité commune de deux noms dont l'un exprime un degré supérieur, comme "أكبر" (plus grand).
Exemple : "أكبرُ سنًّا" (plus âgé).

- Le syntagme quasi-propositionnel reposant sur la qualité similaire (مركب شبه إسنادي قائم على الصّ المشبّهة) : Les noms de la qualité similaire indiquent la présence absolue de la qualité de celui qui a fait l'action, comme "ضعيف" (faible).
  Exemple : "ضعيف العقل" (faible d'esprit).
- Le syntagme quasi-propositionnel reposant sur un nom verbal (مركب شبه إسنادي قائم على المصدر) : Le nom verbal est construit à partir des formes verbales simples ou augmentées combinées avec différents schèmes. Ils correspondent à l'infinitif en français.
  Exemple : "مخالفة له" (contraire à lui).

— Le syntagme prépositionnel (مركب حرفي) : contient obligatoirement une particule appelée mot outil (حرف). Ce dernier peut introduire un syntagme nominal ou se lier à un enclitique. Les mots outils sont : les prépositions (حروف الجر), les outils de stipulation (أدوات الشرط), les outils d'exclusion (أدوات الحصر) et les outils d'exception (أدوات استثناء).
Exemple : في الساحة (dans la cour), إلا الولدَ(sauf le garçon)

— Le syntagme subordonné (مركب موصولي) : contient un outil de liaison (اسم موصول). Les outils de liaisons varient selon le type du syntagme subordonné :

- Les outils de liaison pour un syntagme subordonné nominal sont : من, ما, الذي.
  Exemple : "الذي فاز في السباق" (celui qui a gagné [dans] la course)
- Les outils de liaison pour un syntagme subordonné prépositionnel sont : لو, ما, أن, أنّ, كي.
  Exemple : "كي يفوز في السباق" (pour gagner [dans] la course).

2.2.2.2 Phrase verbale

La phrase verbale sert à indiquer un évènement ou une action. Sa structure est généralement composée comme suit :
[Verbe / فعل] + [Sujet/ فاعل] + [Complément (obligatoire ou optionnel) direct ou indirect/ مفعول به]
Exemple (1) :

ينامُ عليٌ / Dort Ali

Cet exemple est composé d'un verbe intransitif "نَامَ" (dormir) sous sa forme conjuguée à la troisième personne du singulier du masculin "ينامُ" (dort), suivi par son sujet "عليٌ" (Ali) qui est un nom propre.
Le sujet peut être implicite dans le cas où il est déjà exprimé par la forme conjuguée du

verbe. Ce phénomène est appelé sujet elliptique.
Exemple (2) :

نَامَتْ / A dormi

La phrase est constituée du verbe "نَامَ" (dormir) conjugué au passé à la troisième personne "نَامَتْ" (a dormi). La forme conjuguée de ce verbe permet de déduire que son sujet correspond à la troisième personne du singulier et féminin (elle).
La présence du complément est conditionnée par la transitivité du verbe. L'exemple suivant illustre le cas d'un verbe transitif qui a besoin de deux compléments d'objets afin d'obtenir une phrase syntaxiquement correcte.
Exemple (3) :

أعطَى عليٌ الكتابَ إلى فاطمةٍ / A donné Ali le livre à Fatima

La phrase est constituée par le verbe transitif "أعطَى" (donner), un sujet "عليٌ" (Ali) et deux compléments d'objet dont l'un est direct "الكتابَ" (le livre) et l'autre est indirect "إلى فاطمةٍ" (à Fatima). Ce dernier est introduit par une préposition "إلى" (à) qui a un lien étroit avec le verbe. Cette construction correspond au complément d'objet second dans la grammaire du français.
Le complément optionnel (adverbial, circonstanciel, absolu, de raison, etc.) peut occuper n'importe quelle place dans la phrase. Reprenons l'exemple (1) "ينامُ عليٌ" (dort Ali). Nous pouvons ajouter à cette phrase un complément circonstanciel de lieu "في المنزلِ" (dans la maison) dans deux différentes positions.

(4) ينامُ عليٌ في المنزلِ / Dort Ali dans la maison

(5) في المنزلِ ينامُ عليٌ / Dans la maison dort Ali

Dans (4), le complément circonstanciel de lieu est ajouté au début de la phrase (1) pour obtenir la phrase "ينامُ عليٌ في المنزلِ" (Dort Ali dans la maison). Tandis que dans (5) ce complément circonstanciel de lieu est ajouté à la fin de la phrase (1) pour obtenir "في المنزلِ ينامُ عليٌ" (Dans la maison dort Ali). Nous soulignons que cet ajout n'a pas affecté les autres composants de la phrase (1).

2.2.2.3 Phrase nominale

Une phrase nominale sert à indiquer une certaine information (une qualité, une attitude ou un état) appartenant à quelqu'un ou à quelque chose. Elle est constituée de deux éléments principaux sans qu'ils soient liés l'un à l'autre entre-eux par un verbe :
[Thème / مبتدأ] + [Propos / خبر]
Le thème peut être un nom à l'état déterminé (pronom personnel, un pronom démonstratif, nom propre, etc.) ou un syntagme nominal.
Le propos est un nom déterminant le thème et lui servant d'information. Il peut être un adjectif qualificatif indéterminé, un complément circonstanciel, un syntagme nominal etc.
Exemples :

(6) عليٌ في المنزلِ / Ali dans la maison

Cette phrase est constituée d'un thème nom propre "عليٌ" (Ali) et son propos "في المنزلِ" (dans la maison) est un complément circonstanciel de lieu.

(7) الطقسُ جميلٌ / Le temps beau

Cette phrase a un thème construit par syntagme nominal déterminé "الطقسُ" (le temps) et son propos "جميلٌ" (beau) est un adjectif qualificatif.

(8) هذا كتابٌ / Ceci [est un] livre

Le thème de cette phrase est un pronom démonstratif "هذا" (ceci) et son propos est un syntagme nominal simple "كتابٌ" (livre).

(9) هُوَ صديقي المُفضلُ / Lui mon ami préféré

La phrase est composée d'un thème constitué d'un pronom personnel de la troisième personne du singulier "هُوَ" (lui) et son propos est syntagme nominal adjectival "صديقي المُفضلُ" (mon ami préféré).

2.2.2.4 Phrase passive

En arabe, le passage de la forme active d'une phrase verbale vers la forme passive doit respecter un ensemble de règles illustré dans le Tableau 2.2 :

Elément de la phrase verbale	**Forme active**	**Forme passive**
Verbe	S'accorde avec le sujet	S'accorde avec le sujet adjoint
Sujet	C'est le sujet du verbe	C'est le sujet adjoint du verbe
Complément d'objet direct	Son cas est accusatif	Il devient le sujet adjoint du verbe et son cas est nominatif
Complément d'objet indirect	Son cas est génitif	Reste inchangé

TABLE 2.2: Les règles de passage de la forme active vers la forme passive

Prenons l'exemple de la phrase (10) "أَعطَى الطبيبُ المريضةَ الدواءَ" (donne le médecin à la malade le médicament) dans sa forme active. Nous appliquons les règles de passage pour transformer cette phrase sous forme passive.
Tout d'abord le sujet "الطبيبُ" (le médecin) disparait en laissant sa place à son premier complément d'objet "المريضةَ" (la malade). Ce dernier devient donc le sujet adjoint "المريضةُ" et le cas passe de l'accusatif vers le nominatif (de la fatha vers la dhamma). Ainsi, le verbe "أَعطَى" (donner) s'accorde avec son nouveau sujet adjoint, c'est à dire, à la 3ième

personne du singulier au féminin dans sa forme passive "أُعطِيَتْ" (a été donnée). Le complément d'objet second, quand à lui, reste inchangé. Finalement, nous obtenons la forme passive suivante (11) : "أُعطِيَتْ المريضةُ الدواءَ" (a été donnée la malade le médicament).

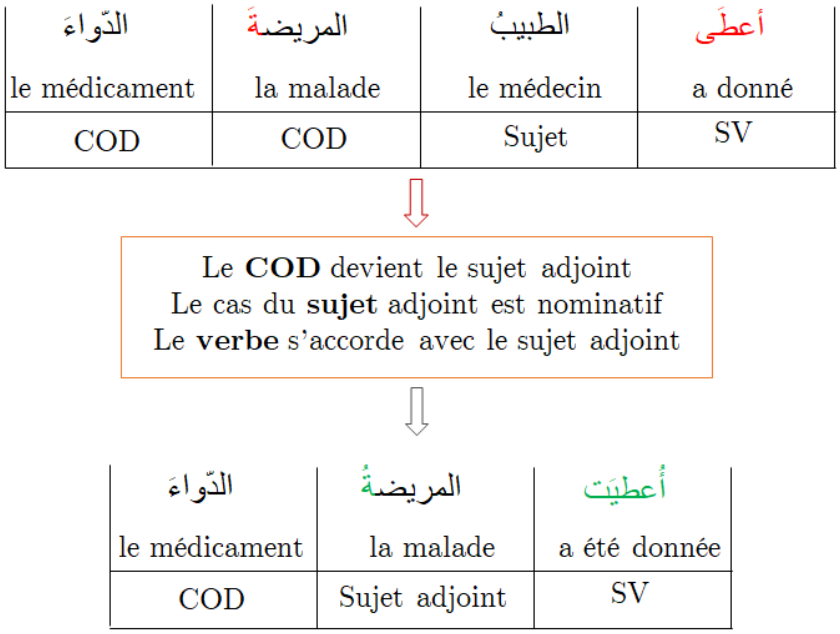


FIGURE 2.2: Exemple de passage de la phrase (Donne le médecin à la malade le médicament) de sa forme active vers sa forme passive

2.2.2.5 Phrase nominale modifiée

Les modifieurs de temps (Alnawasikh / النواسخ) en langue arabe, sont des verbes ou des particules appliqués à la phrase nominale et qui modifient son thème et son propos.

2.2.2.5.a Verbe d'existence

Les verbes d'existence "كان وأخواتها" (Kâna et ses sœurs) introduisent dans la phrase nominale la notion du temps situé, de durée et du devenir. Ils sont rajoutés au début de la phrase nominale et apportent des modifications à son propos. En effet, le cas du thème reste nominatif tandis que le cas du propos change du nominatif à l'accusatif.
Un exemple de cette modification est illustré dans la figure 2.3.
En ajoutant le verbe d'existence "كانَ" (était) au début de la phrase nominale (12), "الطقسُ جميلٌ" (il fait beau) on obtient la phrase "كانَ الطقسُ جميلًا" (il faisait beau). Cet ajout engendre la transformation du cas du propos "جميلٌ" (beau) qui devient accusatif "جميلًا".

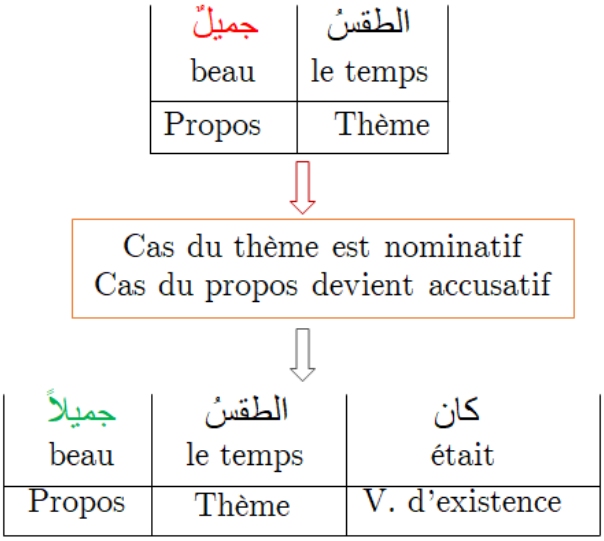


FIGURE 2.3: Exemple d'ajout d'un verbe d'existence au début de la phrase "Il fait beau"

2.2.2.5.b Verbe de certitude

Les verbes de certitude "إن وأخواتها" (Inna et ses sœurs) sont rajoutés au début d'une phrase nominale. Contrairement aux verbes d'existence, ils apportent des changements au thème. Le cas de ce dernier change du nominatif à l'accusatif tandis que le cas du propos reste inchangé.
Un exemple de cette modification est illustré dans la figure 2.4 :

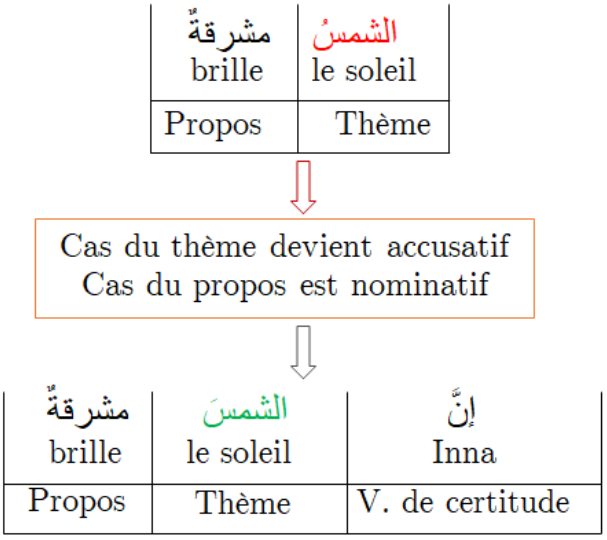


FIGURE 2.4: Exemple d'ajout d'un verbe de certitude au début de la phrase "Le soleil brille"

L'ajout du verbe de certitude "إنَّ" au début de la phrase (13) "الشمسُ مشرقةٌ" (Le

soleil brille) entraîne le changement du cas du thème du nominatif "الشمسُ" (soleil) vers l'accusatif "الشمسَ".

2.2.2.6 Règles d'accord

La construction correcte d'une phrase en langue arabe, doit obéir à un ensemble de règles d'accord.
En phrase verbale, les règles d'accord concernent le verbe et son sujet : Lorsque le verbe précède le sujet, l'accord du verbe se fait seulement en genre et jamais en nombre. Le verbe reste conjugué au singulier. Le tableau 2.3 regroupe des exemples d'accord dans différentes phrases :

Verbe	**Sujet**	**Exemple**
Masculin, Singulier	Masculin, singulier, humain	يَنَامُ الولدُ (dort le garçon)
Féminin, Singulier	Féminin, singulier, humain	تَنامُ البنتُ (dort la fille)
Masculin, Singulier	Masculin, pluriel, humain	يَنَامُ الأولادُ (dort les garçons)
Féminin, Singulier	Féminin, pluriel, humain	تَنامُ البناتُ (dort les filles)
Masculin, Singulier	Masculin, singulier, non humain	يَنَامُ الكلبُ (dort le chien)
Masculin, Singulier	Masculin, dual, non humain	يَنَامُ الكلبَانِ (dort les deux chiens)
Féminin, Singulier	Féminin, pluriel, non humain	تَنامُ الكلابُ (dort les chiens)

Table 2.3: Exemples d'accord entre le verbe et le sujet lorsque le verbe précède le sujet

Lorsque le sujet précède le verbe deux cas sont distingués :

— Si le sujet est un être humain, un non-humain singulier ou encore un non-humain dual, le verbe s'accorde en genre et en nombre.

— Si le sujet est non-humain pluriel, le verbe s'accorde au féminin singulier.

Ces règles d'accord sont illustrées dans le tableau 2.4 :

Verbe	**Sujet**	**Exemple**
Masculin, Singulier	Masculin, singulier, humain	الولدُ يَنَامُ (le garçon dort)
Féminin, Singulier	Féminin, singulier, humain	البنتُ تَنامُ (la fille dort)
Masculin, pluriel	Masculin, pluriel, humain	الأولادُ ينامونَ (Les garçons dorment)
Féminin, pluriel	Féminin, pluriel, humain	البناتُ يَنمنَ (les filles dorment)
Masculin, Singulier	Masculin, singulier, non humain	الكلبُ يَنَامُ (le chien dort)

Masculin, dual	Masculin, dual, non humain	الكلبَانِ ينَامَان (les deux chiens dorment)
Féminin, Singulier	Féminin, pluriel, non humain	الكلابُ تَنامُ (les chiens dort[dorment])

TABLE 2.4: Exemples d'accord entre le verbe et le sujet lorsque le sujet précède le verbe

Dans un syntagme adjectival, l'accord entre l'adjectif et le nom qu'il qualifie se fait comme suit :

— Si le nom qualifié est un être humain, non-humain singulier ou bien non-humain dual, l'accord de l'adjectif se fait en genre, en nombre, en cas et en détermination.

— Si le nom qualifié est un non-humain pluriel, l'adjectif est au féminin singulier et l'accord se fait en cas et en détermination.

Le tableau 2.5 présente un ensemble d'exemples qui couvrent ces règles d'accord :

Nom qualifié	**Adjectif**	**Exemple**
Masculin, singulier, humain, déterminé	Masculin, singulier, humain, déterminé	الطالب النجيب (l'étudiant l'excellent)
Féminin, singulier, humain, déterminé	Féminin, singulier, humain, déterminé	الطالبة النجيبة (l'étudiante l'excellente)
Masculin, dual, humain, déterminé	Masculin, dual, humain, déterminé	الطالبين النجيبان (les deux étudiants les excellents)
Féminin, dual, humain, déterminé	Féminin, dual, humain, déterminé	الطالبتان النجيبتان (les deux étudiantes les excellentes)
Masculin, pluriel, humain, déterminé	Masculin, pluriel, humain, déterminé	الطلبة النجباء (les étudiants les excellents)
Féminin, pluriel, humain, déterminé	Féminin, pluriel, humain, déterminé	الطلبات النجيبات (les étudiantes les excellentes)
Féminin, pluriel, non humain, déterminé	Féminin, singulier, humain, déterminé	الكلاب الجائعة (Les chiens l'affamée[affamés])

TABLE 2.5: Exemples d'accord entre l'adjectif et le nom qualifié dans un syntagme adjectival

Dans une phrase nominale, l'accord entre le thème et le propos, lorsque ce dernier est un adjectif, se fait exactement comme pour le syntagme adjectival. Cependant, le propos ne s'accorde pas en cas ni en détermination avec son thème.
Ceci est illustré par les exemples du tableau 2.6 :

Thème	**Propos (Adjectif)**	**Exemple**
Masculin, singulier, humain, déterminé	Masculin, singulier, humain, non déterminé	الطالب نجيب (l'étudiant excellent)
Féminin, singulier, humain, déterminé	Féminin, singulier, humain, non déterminé	الطالبة نجيبة (l'étudiante excellente)
Masculin, dual, humain, déterminé	Masculin, dual, humain, non déterminé	الطالبين نجيبان (les deux étudiants excellents)
Féminin, dual, humain, déterminé	Féminin, dual, humain, non déterminé	الطالبتان نجيبتان (les deux étudiantes excellentes)
Masculin, pluriel, humain, déterminé	Masculin, pluriel, humain, non déterminé	الطلبة نجباء (les étudiants excellents)
Féminin, pluriel, humain, déterminé	Féminin, pluriel, humain, non déterminé	الطلبات نجيبات (les étudiantes excellentes)
Féminin, pluriel, non humain, déterminé	Féminin, singulier, humain, non déterminé	الكلاب جائعة (Les chiens affamée [affamés])

TABLE 2.6: Exemples d'accord entre le thème et le propos dans une phrase nominale

2.3 Difficultés de traitement morphosyntaxique de la langue arabe

En langue arabe, la morphologie et la syntaxe sont intimement liées puisque la morphologie peut exprimer des relations syntaxiques. Par exemple, les sujets des verbes ont un cas nominatif et les modificateurs adjectivaux des noms ont le même cas que le nom qu'ils modifient.
Nous nous focalisons dans cette section sur les caractéristiques morphosyntaxiques qui rendent la tâche d'analyse syntaxique difficile à mettre en œuvre.

2.3.1 Voyellation

La majorité des documents arabes sont non voyellés. Les voyelles ne sont utilisées que pour certains ouvrages scolaires pour débutants et pour le Coran. L'absence de ces voyelles peut donc engendrer beaucoup de cas ambigus au cours de l'analyse d'un texte et empêcher son traitement correct. Dans ce cas, seul le contexte permettra de comprendre le sens de la phrase.
Prenons l'exemple de la forme non voyellée suivante : "كتب" (écrire). Cette forme peut accepter les voyellations suivantes :

Racine	Voyellations possibles
كتب	كَتَبَ (il a écrit)
	كَتَّبَ (il a fait écrire)
	كَتْب (rédaction)
	كُتَبٌ (des collections)
	كُتُبٌ (des livres)

TABLE 2.7: Exemple de l'ambiguïté vocalique du mot écrire

Nous pouvons remarquer que chacune de ces voyellations peut engendrer une (ou plusieurs) catégorie(s) grammaticale(s). D'où la deuxième spécificité de la langue arabe qu'est l'ambiguïté grammaticale.

2.3.2 Ambiguïté grammaticale

Les mots en langue arabe sont relativement plus ambigus grammaticalement que pour les langues latines. En effet, un mot peut avoir plus d'une valeur grammaticale. Les statistiques réalisées par [Ben Othmane Zribi, 1998] confirment que le taux d'ambiguïté grammaticale pour les formes lexicales augmente en l'absence des signes vocaliques et aussi à cause de l'agglutination des enclinomènes aux formes textuelles simples.
Le tableau 2.8 regroupe certaines de ses différentes formes voyellées avec leurs valeurs grammaticales respectives du mot "كتب" (écrire).

Formes	Catégories grammaticales correspondantes
كَتَبَ (il a écrit)	Verbe à la 3ème personne du masculin singulier à l'accompli actif
كَتَّبَ (il a fait écrire)	Verbe à la 3ème personne du masculin singulier à l'accompli actif
كَتْب (rédaction)	Nom verbal du verbe écrire
كُتَبٌ (des collections)	Nom pluriel de collection
كُتُبٌ (des livres)	Nom pluriel de livre

TABLE 2.8: Exemple de l'ambiguïté grammaticale des formes textuelles du mot écrire

Sans voyellation, ce mot "كتب" (écrire) peut donc se référer à un verbe (écrire) ou encore à un nom (livres).

2.3.3 Agglutination

L'agglutination des enclinomènes engendre des ambiguïtés morphologiques et syntaxiques au cours de l'analyse. Pour analyser syntaxiquement une forme agglutinée constituant une proposition, il faut procéder à son découpage en proclitique/radical/enclitique. Ce découpage est lui-même confronté à un problème d'ambiguïté vu que pour une seule unité lexicale on peut avoir plusieurs découpages possibles.
Par exemple, la forme de surface "فأبه" (être intéressé) peut être découpée en triplets proclitique/radical/enclitique de différentes manières telles que par exemple : ف / أبه (être intéressé) ou ف / أب / ه (l'aborder).
De plus, il est difficile de distinguer entre un proclitique ou enclitique et un caractère du mot en question. Par exemple, le caractère "و" fait partie du mot "وَعَدَ" (promettre) mais il s'agit d'un proclitique dans le mot "وكتب" (et il a écrit).

2.3.4 Ordre semi-libre des mots

L'ordre des mots en arabe est relativement libre. Ainsi, on peut, par exemple, changer l'ordre des mots dans une phrase verbale composée d'un Verbe (V), d'un Sujet (S) et d'un Objet (O). Il est à noter que l'ordre le plus utilisé dans l'Arabe standard est le VSO. Le changement de l'ordre des composants d'une même phrase peut engendrer aussi un changement du type de la phrase qui devient une phrase nominale.
Prenons l'exemple d'une phrase verbale avec l'ordre standard VSO (17) : "ولدَ الباحثُ فِي تونس" (est né le chercheur en Tunisie) illustré par la figure 2.5.

في تونس en Tunisie	الباحث le chercheur	ولد est né
COI	Sujet	V

في تونس en Tunisie	ولد est né	الباحث le chercheur
COI	V	Sujet

الباحث le chercheur	ولد est né	في تونس en Tunisie
Sujet	V	COI

FIGURE 2.5: Changement de l'ordre des mots de la phrase "Est né le chercheur en Tunisie"

En permutant les composants de cette phrase, ces deux autres combinaisons sont obtenues : "الباحثُ ولدَ

"فِي تونس" (le chercheur est né en Tunisie) (SVO) et "فِي تونس ولدَ الباحثُ" (en Tunisie est né le chercheur) (OVS) ayant le même sens.
Cependant, lorsqu'un complément d'objet direct se place devant le verbe, ce dernier se verra rajouter une anaphore correspondante au complément d'objet. Ce phénomène est illustré par la figure 2.6.

الكرة	اللاعب	ركل
le ballon	le joueur	a frappé
Sujet	COD	SV

Phrase verbale

اللاعب	ركلها	الكرة
le joueur	l'a frappée	le ballon
Sujet	V + anaphore	COD

Phrase nominale

FIGURE 2.6: Ajout d'une anaphore au complément d'objet qui précède un verbe

Lorsque le complément d'objet direct "الكرةَ" (ballon) de la phrase (14) : "ركلَ اللّاعبُ الكرةَ" (a frappé le joueur le ballon) change de position et devance le verbe, l'anaphore qui correspond à ce complément "ها" (elle) est attachée à la fin du verbe "ركلَها".
Ce problème ne se pose pas dans le cas d'un complément d'objet indirect (syntagme prépositionnel) qui se place dans n'importe quelle position dans la phrase sans contrainte.

للاّعبين	الجمهور	صفّق
aux joueurs	le public	a applaudi
Sujet	COD	SV

Phrase verbale

الجمهور	صفّق	للاّعبين
le public	a applaudi	aux joueurs
Sujet	V	COD

Phrase nominale

FIGURE 2.7: Exemple du changement de la position du complément d'objet directe dans la phrase "Le public a applaudi les joueurs"

Dans la phrase (15) "صفّقَ الجمهورُ للّاعبين" (le public a applaudi les joueurs), illustrée par la figure 2.7, le complément d'objet indirect "للّاعبين"(aux joueurs), qui est un syntagme prépositionnel (introduit par la proposition "ل"), peut se placer à n'importe quelle position dans la phrase, sans contrainte.
L'ordre semi-fixe ne concerne pas seulement les composantes essentielles de la phrase mais touche aussi ses compléments (les circonstanciels de temps, de lieu, les compléments de manière, etc.). Par conséquence, il faudra prévoir dans la grammaire toutes les règles de combinaisons possibles et d'inversion de l'ordre des mots dans la phrase.

2.3.5 Segmentation des textes

La segmentation est une phase importante pour l'analyse syntaxique. Elle vise à délimiter les frontières des phrases et permet ainsi de découper le texte et repérer les segments contenant les informations recherchées.
Le problème de segmentation pour la langue arabe est que, contrairement aux langues latines, elle ne s'appuie pas principalement sur les signes de ponctuations. Ainsi il est fréquent qu'un paragraphe ne contient aucun signe de ponctuation à part le point marquant sa fin. On parle alors de macro-phrases. Ce phénomène s'explique en raison des liaisons entre les phrases moyennant les conjonctions de coordinations. Considérons l'exemple suivant (16) :

اِشتَدَّ ضَربُ الطُّبولِ وتَعَالتْ أصواتُ المَزاميرِ ثمَّ توالتِ الطلقاتُ الناريةُ

Les coups de tambour se sont intensifiés et les sons des flûtes se sont fait entendre, puis les coups de feu ont été tirés.

Cette macro-phrase contient deux conjonctions de coordinations "و"(et) et "ثمَّ" (puis). Par conséquent, nous pouvons la décomposer en ces trois micro-phrases :

[اِشتَدَّ ضَربُ الطُّبولِ][وتَعَالتْ أصواتُ المَزاميرِ][ثمَّ توالتِ الطلقاتُ الناريةُ]

[Les coups de tambour se sont intensifiés] [et les sons des flûtes se sont fait entendre] [puis les coups de feu ont été tirés].

2.3.6 Enchâssement

Les structures récursives ou enchâssées, communément connues sous le nom de subordonnées relatives, sont très fréquentes dans les textes en langue arabe. Ce genre de structures peut être illustré par l'exemple suivant :
(17) "المدير سلّم الجائزة للتلميذ" (le directeur a remis le prix à l'élève)
(18) "المدير هو الذي سلّم الجائزة للتلميذ" (C'est le directeur [qui a remis le prix] à l'élève)
(19) "المدير هو الذي سلّم الجائزة للتلميذ الذي حقق النجاح" (C'est le directeur qui a remis le prix à l'élève [qui a réussi])
Par conséquent, la longueur d'une phrase n'est pas limitée et la segmentation en micro-phrases n'est pas permise dans ce cas (pas de conjonctions de coordinations).

2.3.7 Interprétation syntaxique

Ce phénomène existe dans différentes langues naturelles tels que l'anglais le français et aussi l'arabe. Une phrase ou un extrait de texte mis dans un contexte bien spécifique peut être interprété syntaxiquement de différentes manières. L'entrelacement entre syntaxe et sémantique et les problèmes que nous avons recensés dans les paragraphes précédents peuvent mener à cette multiplicité des interprétations syntaxiques. Comme exemples de phrases ayant plus d'un arbre syntaxique possible, nous présentons la phrase suivante (figure 2.8).

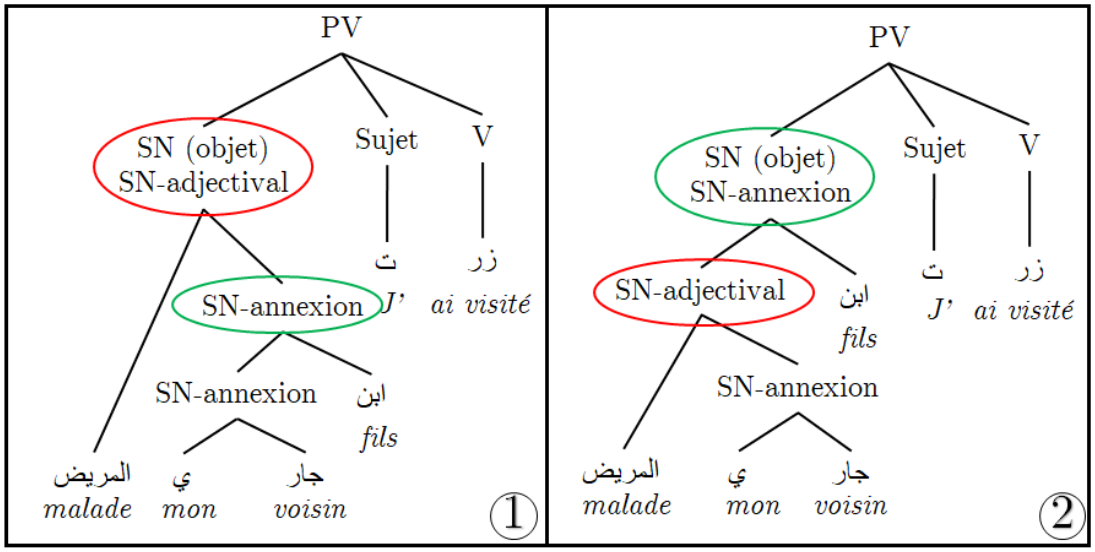


FIGURE 2.8: Deux arbres syntaxiques différents pour représenter la phrase "J'ai visité le fils de mon voisin malade"

La phrase verbale (20) "زرت ابن جاري المريض" (J'ai visité le fils de mon voisin malade) peut admettre deux structures syntaxiques représentées par la figure ci-dessus. Dans la première structure, l'adjectif "المريض" (le malade) a été associé au syntagme d'annexion "ابن جاري" (fils de mon voisin). Dans la seconde, nous remarquons que le même adjectif "المريض" (le malade) qualifie la forme agglutinée "جاري" (mon voisin). Donc, suivant ce que cet adjectif qualifie dans la phrase, la structure syntaxique affectée change ainsi que le sens de la phrase.

2.4 Ressources d'analyse de la langue arabe

Pour une analyse fiable d'une langue naturelle, il est indispensable d'avoir à disposition un ensemble de ressources pour son traitement. La qualité des résultats, produits par les applications de TALN, est liée à la qualité de ces ressources en termes de nombre d'entrées et d'exhaustivité des informations qu'elles contiennent. Par exemple, lorsque nous utilisons un dictionnaire électronique pour une analyse morphologique, les mots (corrects) peuvent être considérés incorrects s'ils ne figurent pas dans le dictionnaire utilisé. Par conséquent,

il est nécessaire d'avoir à disposition des ressources numériques, pour le traitement de l'arabe, adaptées et de haute qualité, en matière de contenu et de structure.
Nous avons exploré certaines de ces ressources, que nous présentons dans cette section, traitant de la syntaxe et la sémantique de l'arabe sujet de notre étude.

2.4.1 Ressources syntaxiques

Les corpus arborés (Treebanks) représentent un ensemble de phrases analysées (vérifiées manuellement) et stockées sous forme d'arbres. En TALN, ces corpus constituent une ressource importante pour la construction d'analyseurs syntaxiques à base d'apprentissage et leur évaluation. Ils sont également utilisés pour analyser automatiquement des corpus de taille importante et pour diverses applications telles que la diacritisation, l'étiquetage morphosyntaxique, la segmentation de la phrase, l'étiquetage sémantique. La création d'un corpus arboré est confrontée à un compromis entre la richesse linguistique et la taille du corpus. Plus l'annotation est riche, plus le processus d'annotation est lent et la taille du corpus arboré est réduite. Par conséquence, la disponibilité de ces ressources diffère selon les langues.
Pour l'arabe standard, il existe trois corpus arborés importants : Penn Arabic Tree-Bank (PATB) [Maamouri and Bies, 2004], Prague Arabic Dependency Treebank (PADT) [Hajič et al., 2004] et Columbia Arabic Treebank (CATiB) [Habash and Roth, 2009]. Dans cette section, nous donnons une brève présentation de chacune de ces trois ressources.

2.4.1.1 Penn Arabic Treebank

Le projet Penn Arabic Treebank (PATB) a démarré en 2001 avec le consortium de données linguistiques (Linguistic Data Consortium : LDC) et à l'Université de Pennsylvanie (lieu de naissance des Treebank pour l'anglais [Marcus et al., 1993], chinois [Xue et al., 2002] et coréen [Han et al., 2002]). A ce jour, quatre parties du PATB ont été mis en place [Maamouri et al., 2003] ; [Maamouri et al., 004a] ; [Maamouri et al., 004b] ; [Maamouri et al., 2005]. Chacune de ces parties a été publiée dans différentes versions avec différents degrés d'améliorations. Ce corpus est constitué de textes non voyellés extraits d'articles de différentes agences de presse (France Presse, Xin-hua, Al-Hayat, Ummatée ah Press, etc.). La plupart des phrases de ces textes ont été traduites en anglais ou sont associées à des traductions.
Chaque occurrence d'un mot dans PATB est étiquetée avec sa catégorie grammaticale et ses fonctions. Le mot est ainsi marqué morphologiquement, syntaxiquement, et avec d'autres relations pertinentes telles la concordance et l'accord. PATB, inclut plus de 400 étiquettes POS différentes. Ces dernières offrent les informations morphosyntaxiques pour le cas, le mode, le genre et la définition [Maamouri et al., 2010]. Parmi ces étiquettes, 22 sont syntagmatiques (des catégories syntaxiques), 20 sont des relations syntaxiques et sémantiques et 24 représentent les étiquettes POS de base.
Un exemple d'arborescence de PATB est illustré par la figure 2.9. L'arbre entier représente la phrase (21) "خمسون ألف سائح زاروا لبنان و سوريا في أيلول الماضي" (Cinquante mille touristes ont visité le Liban et la Syrie en septembre dernier) étiquetée par la catégorie S. Cet arbre est composé de deux syntagmes de mots : un syntagme nominal, étiqueté NP-TPC, servant de sujet et un syntagme verbal étiqueté VP. Le sujet est constitué d'un

chiffre suivi d'un syntagme nominal (NP). Ce dernier est constitué aussi d'un chiffre suivi d'un syntagme nominal (NP). Ce NP contient un seul nom. Dans PATB la configuration (NP NOUN NP) est utilisée pour marquer les constructions Idafa (annexion) et Tamyiz [4] (le spécificatif). Les deux constructions ne peuvent être distinguées qu'en utilisant le cas morphologique du nom dans le NP incorporé. Le syntagme verbal (VP) est composé de sept mots de la phrase Il contient un sujet vide (NP-SBJ) indiquant qu'il est pronominal et conjugué au verbe, un objet (NP-OBJ) et un syntagme prépositionnel temporel (PPTMP).

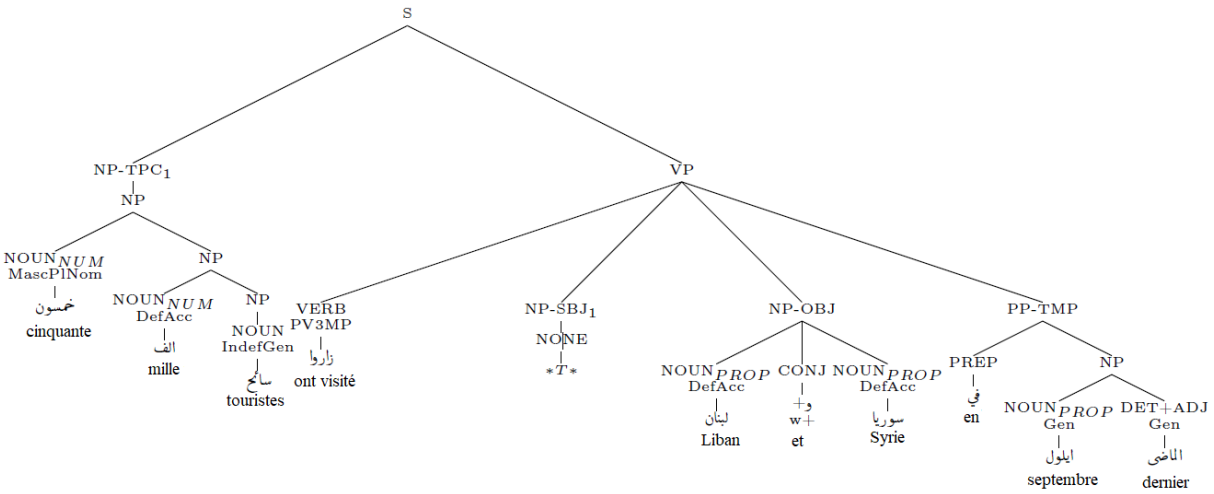


FIGURE 2.9: Représentation de la phrase "Cinquante mille touristes ont visité le Liban et la Syrie en septembre dernier" en structure de syntagmes dans PATB

2.4.1.2 Prague Arabic Dependency Treebank

Prague Arabic Dependency Treebank (PADT) [Hajič et al., 2004] a été généré à l'Institut de linguistique formelle et appliquée de l'Université Charles de Prague. Ce corpus contient une description de la phrase à plusieurs niveaux : la morphologie fonctionnelle, la syntaxe de dépendance analytique et la représentation tectogrammatique [5] (pour la constituante et les fonctions) de la signification linguistique. Ces annotations linguistiques sont basées sur la théorie de la description générative fonctionnelle (FGD) [Sgall et al., 1986] ainsi que le Prague Dependency Treebank [Hajič et al., 2001].
Ce corpus a été crée à base du PATB. Ce dernier a été converti vers les représentations syntaxiques du PADT en plus d'autres textes (provenant des articles de presse de Xinhua, Al-Hayat, An-Nahar) qui ont été annotés. En effet, les annotations morphologiques et syntaxiques dans PADT diffèrent considérablement de PATB. Les annotations POS figurent dans un ensemble d'étiquettes morphologiques développées dans ElixirFM [Smrž, 2007]. [6]

4. C'est un substantif indéfini et mis à l'accusatif. Il est placé immédiatement après la préposition dont il limite ou définit l'attribut.

5. La structure tectogrammaticale reflète la manière dont la phrase est composée à partir de ses parties, elle définit les fonctions grammaticales (sujet, objet, etc.), et c'est cette structure qui est en interface directe avec la sémantique.

6. ElixirFM est une implémentation de haut niveau de la morphologie arabe fonctionnelle.

Et contrairement à PATB qui utilise la structure à base de constituants, PADT adopte la structure de dépendances. Les structures de dépendances prennent également la forme d'arbres sauf que les mots de la phrase sont les nœuds de l'arbre. En plus, plusieurs types de relations, entre les nœuds, sont spécifiés.

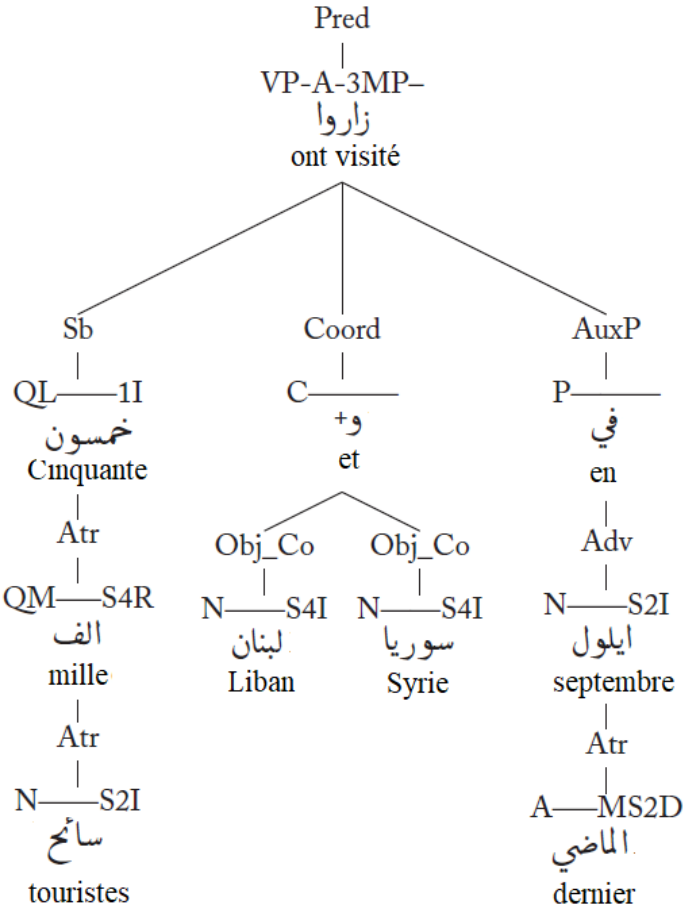


FIGURE 2.10: Représentation de la phrase "Cinquante mille touristes ont visité le Liban et la Syrie en septembre dernier" dans PADT

La figure 2.10 illustre les représentations en structures de dépendances de la même phrase (21) dans PADT. Dans cette représentation, la tête de la phrase (VP-A-3MP) est le verbe "زاروا" (ont visité) qui est la forme active conjugué à la 3ième personne du pluriel masculin de "زار" (visiter). Ce nœud a trois nœuds fils, un sujet (Sb), une conjonction de coordination (Coord) et un syntagme prépositionnel auxiliaire (AuxP). Le sujet contient un mot numérique modifié par un autre mot numérique dans une relation attributive. Ce dernier est également modifié par un autre mot dans une relation attributive. Le deuxième nœud fils du verbe, relie les deux noms propres avec la relation de composition Obj_Co. Obj_Co indique que les deux noms propres sont coordonnés (Co) par leur parent et qu'ils sont tous deux des objets (Obj) de leur verbe. En ce qui concerne le dernier nœud fils du verbe, la préposition "في" (en) est en relation adverbiale (adv) avec le nom du mois "Septembre", qui dirige un adjectif dans une relation attributaire (Atr).

2.4.1.3 Columbia Arabic TreeBank

Le projet Columbia Arabic TreeBank (CATiB) a débuté à l'université de Columbia en 2008. Les textes annotés de ce corpus proviennent des articles de presse (provenant de France Presse, Xinhua, Al-Hayat, Al-Asharq Al-Awsat, Al-Quds Al-Arabi, An-Nahar, Al-Ahram et As-Sabah) ainsi que ceux convertis à partir de PATB. CATiB contraste avec les autres approches de mise en place des TreeBank en mettant l'accent sur une production plus rapide avec quelques contraintes sur la richesse linguistique [Habash and Roth, 2009]. En effet, CATiB évite l'annotation d'informations linguistiques redondantes. Par exemple, les cas sont déterminés automatiquement à partir de la syntaxe et de l'analyse morphologique des mots et donc, ils peuvent ne pas être annotés au préalable. De plus, CATiB utilise une représentation et terminologie intuitives des dépendances prises de la syntaxe arabe traditionnelle, tels que les composants d'annexion, etc.
Dans ce corpus, il existe huit relations syntaxiques utilisées pour étiqueter les dépendances : sujet (SBJ), objet (OBJ), prédicat (PRD), topic (TPC), Idafa (IDF), Tamyiz (TMZ), modificateur (MOD), et plat (-).
Reprenons l'exemple de la phrase (21). La figure 2.11 illustre sa représentation en structures de dépendances dans CATiB.

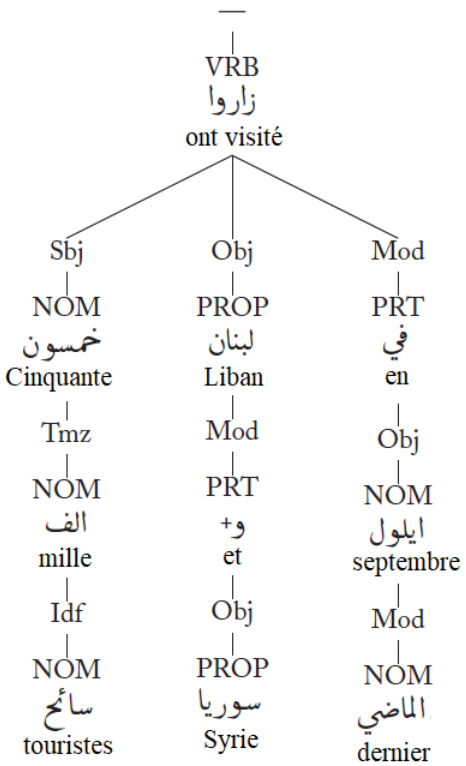


FIGURE 2.11: Représentation de la phrase "Cinquante mille touristes ont visité le Liban et la Syrie en septembre dernier" dans CATiB

Nous pouvons remarquer que la représentation des dépendances est similaire à celle utilisée dans PADT mais avec quelques différences. La tête de la phrase (VRB) est le verbe "زاروا" (ont visité). Il a trois nœuds fils, un sujet (Sbj), un objet (Obj) et un modificateur

prépositionnel (MOD). Le sujet contient une expression complexe du nombre contenant une relation entre Idafa et Tamyiz. Quant à l'objet, il définit la coordination de deux noms avec une particule de conjonction. Le troisième nœud fils du verbe, débute par la préposition liée à un objet (OBJ), lui-même modifié par un nominal adjectival. Cette simplicité des relations utilisées, constitue le signe distinctif de l'annotation CATiB par rapport aux autres approches d'arborescence.

2.4.2 Ressources sémantiques

Afin de déterminer le sens approprié d'un mot ambigu dans un texte selon un contexte précis, il est primordial de disposer d'un ensemble de ressources sémantiques. Ces dernières permettent une meilleure compréhension et une meilleure exploitation des contenus textuels pour diverses applications de TALN, telles que la traduction automatique, la recherche d'information, la classification, etc. Nous présentons deux des plus importantes ressources traitant de la sémantique de l'arabe.

2.4.2.1 Arabic PropBank

La Proposition Bank (ou PropBank) [Palmer et al., 2005] est un corpus annoté sémantiquement. Elle ajoute sur le Penn Treebank une annotation des structures sous forme de prédicat-argument avec étiquetage en rôles sémantiques des arguments. La version arabe de PropBank, Arabic PropBank, a été mise en place par l'Université du Colorado. Elle adopte une approche similaire à celle utilisée pour le développement de la langue anglaise. Arabic PropBank repose sur la structure syntaxique de Penn Arabic TreeBank (PATB) et aux annotations de lemmes présents dans cette dernière [Palmer et al., 2008].
L'objectif de PropBank est de fournir des étiquettes d'arguments cohérentes dans différentes réalisations syntaxiques du même verbe comme dans l'exemple suivant :

(22) قرأَ [arg0 عليٌ] [arg1 الروايةَ] / Ali lit le roman

(23) قُرِئت [arg1 الروايةُ] / Le roman a été lu

Les arguments des verbes sont étiquetés comme des arguments numérotés : Arg0, Arg1, Arg2 etc. à chaque verbe, des rôles sémantiques sont spécifiés. Les étiquettes permettent d'annoter un même rôle sémantique dans toutes les variations syntaxiques d'un même verbe. Par exemple, le verbe "قرأَ" (lire) dans sa forme active de la phrase (22) admets deux arguments : arg0 pour "عليٌ" (Ali) et arg1 pour "الروايةَ" (roman) alors que dans l'exemple (23) le verbe "قرأَ" (lire) sous sa forme passive admet un seul argument arg1 pour "الروايةَ" (roman).
En plus de ces rôles spécifiques à chaque verbe, PropBank définit un ensemble de rôles génériques appelés Argm. Ces rôles correspondent aux ajouts tel que le lieu, le temps, la durée, la cause, etc. Plus précisément, dans Arabic PropBank 24 types d'arguments sont définis : cinq arguments principaux numérotés (Arg0, Arg1, Arg2, Arg3, Arg4) et 19 arguments complémentaires, qui comprennent les Argm.
Arabic PropBank fournit donc un lexique qui rassemble des framesets (ensemble de cadres

sémantiques). Ces cadres sémantiques fournissent une description spécifique à chaque verbe de tous les rôles sémantiques possibles et illustrent ces rôles à l'aide d'exemples. Prenons l'exemple du verbe "قرأَ" (lire). Son frameset, dans Arabic PropBank, contient l'annotation suivante :
Roleset id : 01 , to read
Arg0 : entity reading (Lecteur)
Arg1 : thing being read (chose en cours de lecture)
Arg2 : topic (sujet)
Arg3 : hearer (auditeur)
Exemple (24) : " ما يسمعه اللبناني او يقرأه عن الوقائع الاقتصادية المالية " (Ce que le Libanais entend ou lit sur les réalités économiques et financières)
Frame :
Arg0 : * > اللبناني
Gloss : -NONE- > the Lebanese
Arg1 : ه *T*-2
Gloss : it
Arg2 : *RNR*-4 > عن الوقائع الاقتصادية و المالية
Gloss : about the economic and financial events
L'annotation, au sein de ce frameset, permet d'établir le lien entre les rôles du Roleset et les arguments dans Frame de la phrase (24). Ainsi, nous pouvons déduire que le lecteur c'est le libanais "اللبناني", la chose en cours de lecture est représentée par le pronom personnel lui "ه" et le sujet concerne les événements économiques et financiers "عن الوقائع الاقتصادية و المالية".

2.4.2.2 Arabic VerbNet

VerbNet [Kipper et al., 2008] est une ressource lexicale pour les verbes anglais. Elle repose sur le système de classification sémantico-syntaxique des verbes de [Levin, 1993]. Les verbes ayant un comportement syntaxique et sémantique similaire sont affectés à un même groupe de classes. Un groupe de classe représente une hiérarchie établie par les relations sémantiques entre ses classes. Chaque classe d'un verbe est décrite au moyen des éléments suivants :

— Les membres : la liste des verbes appartenant à cette classe ou à une sous-classe.
— Les rôles : ce sont les rôles thématiques attribués à chaque membre du verbe de la classe. Ces rôles peuvent admettre un ensemble de restrictions sur leurs natures (animation, location, etc.).
— Les cadres : ils définissent la correspondance entre les rôles sémantiques et les arguments syntaxiques. Pour chaque exemple de phrase, sa structure syntaxique et sa structure sémantique contenant des prédicats sémantiques et leurs arguments sont définis.

Une version arabe de VerbNet appelée "ArabicVerbNet" [Mousser, 2010] ; [Mousser, 2011] a été développée. Elle couvre les verbes les plus utilisés de l'arabe moderne standard. L'organisation des classes de verbes est telle qu'établie par Levin et la procédure de

développement de base de Kipper mais avec quelques adaptations. La version actuelle d'ArabicVerbNet comporte 334 classes qui contiennent 7672 verbes et 1393 cadres.
Les classes fournissent des informations sur la racine verbale, la forme déverbale et le participe des verbes (membres) appartenant à la même classe. Les rôles thématiques sont décrits (avec éventuellement les contraintes sur les rôles), suivis par un ensemble de descriptions syntaxiques (avec exemple de phrase) et les relations sémantiques entre les arguments du verbe.

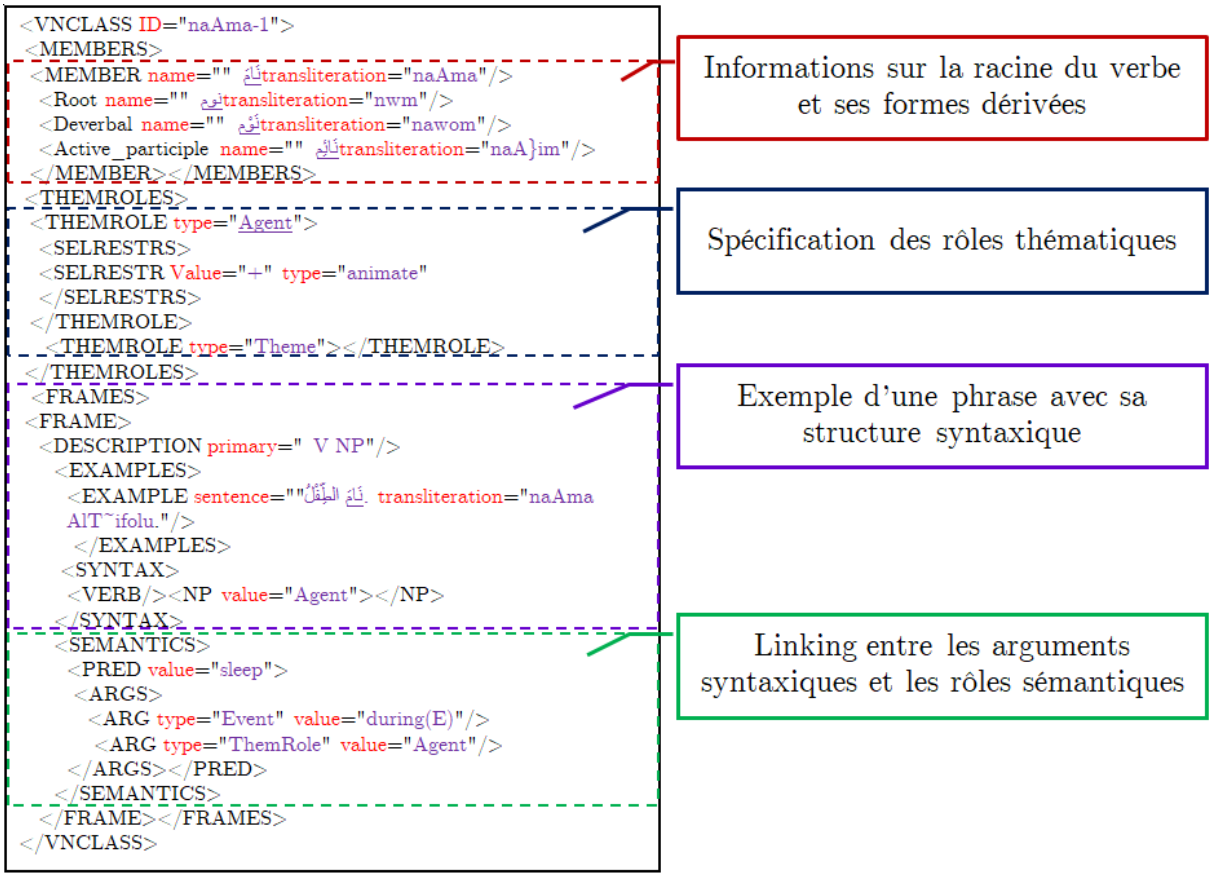


FIGURE 2.12: Extrait du fichier XML de la classe naAma (dormir) dans ArabicVerbNet

Par exemple, pour le verbe "نَامَ" (dormir) de la classe "naAma" (dormir) illustrée par la figure 2.12, sa racine est "نوم", sa forme déverbale est "نَوْم" et son participe actif est "نَائِم". Nous distinguons deux rôles thématiques : le rôle "Agent" qui admet une contrainte sur son type (doit être animé) et le rôle "Thème" sans contrainte particulière. Dans cet extrait de la classe "naAma", une seule description syntaxique est présentée avec un exemple de phrase. Il s'agit de la structure VPN (Verbe + Syntagme nominal) et la phrase exemple de "نَامَ الطِّفْلُ" (l'enfant dort). À la fin, le rôle sémantique affecté au seul argument du verbe de cette structure (le syntagme nominale) et préciser par sa valeur "Agent".

2.5 Grammaires d'arbres adjoints pour la langue arabe

La tâche d'analyse syntaxique et/ou sémantique, requiert non seulement des ressources lexicales mais aussi un ensemble de connaissances et de ressources fournissant des informations de qualité sur les représentations correctes des données d'entrée (texte ou phrase). En effet, le processus d'analyse doit être conforme aux règles d'une grammaire formelle. À l'instar des autres langues naturelles, des grammaires formelles pour l'arabe ont été développées.

Plusieurs méthodes d'analyse de la langue arabe ont adopté l'approche basée sur des règles. Cette approche consiste à utiliser des grammaires formelles bien définies pour représenter la syntaxe arabe. Parmi ceux-ci l'analyseur MASPAR [Belguith et al., 2007], la plateforme PHARAS [Loukam and Laskri, 2008] et l'analyseur de [Haddar et al., 2009] proposent une analyse syntaxique basée sur le formalisme HPSG. [Haddar et al., 2010] ainsi que [Boukedi and Haddar, 2014]ont construit une HPSG. D'autres [Attia, 2008] ont développé une grammaire LFG pour analyser l'arabe. Quant à [Al-Bataineh and Bataineh, 2009] and [Al-Taani et al., 2012] ont utilisé une grammaire sans contexte (CFG) et [Othman et al., 2003] une grammaire basée sur l'unification (UBG). Quant à [Hammouda and Haddar, 2017],ils ont établi une méthode d'analyse syntaxique pour les phrases nominales basée sur un ensemble de règles lexicales et syntaxiques. L'analyseur proposé intègre un processus de désambiguïsation et annote automatiquement les corpus arabes. Toutes ces grammaires traitent uniquement de la syntaxe de l'arabe.
Du coté des TAGs, à notre connaissance, il existe deux travaux qui ont construit une grammaire d'arbres adjoints pour l'arabe. La première est à base de corpus. En effet, [Habash and Rambow, 2004] ont construit une TAG par extraction d'arbres élémentaires à partir de la partie 1 de la version 2.0 de PATB [Maamouri et al., 2003]. Cette extraction passe par une réinterprétation du corpus en des structures de dépendances. Au cours du processus, les structures obtenues sont des structures avec variation des positions de leurs composants : des phrases à composition VSO, d'autres SVO et d'autres OVS. Par contre, les phrases à composition VOS n'ont pas été obtenues. Cet échec est dû à l'absence de ce genre de structures dans le corpus utilisé pour l'extraction. C'est d'ailleurs le problème majeur des approches à base de corpus, vu qu'elles restent limitées aux informations définies dans ces corpus. Comme perspectives, les chercheurs ont voulu continuer à expérimenter ces paramètres d'extraction sur des corpus plus grands afin d'obtenir une grammaire plus exhaustive.

La deuxième est ArabTAG (Arabic Tree Adjoining Grammar) construite au sein de notre laboratoire RIADI [7]-GDL à l'ENSI [8] La Manouba en Tunisie [Ben Fraj, 2010]. C'est une grammaire représentative de la syntaxe arabe. Elle décrit les différents composants syntaxiques de différents niveaux (phrase, syntagme ou macro-catégorie) ainsi que les informations qui leurs sont relatives (morphologiques et syntaxiques). L'étude linguistique de la syntaxe a été réalisée en se référant à des livres scolaires (8ème et 9ème de l'enseignement de base en Tunisie) et aussi au livre de la grammaire arabe de [Kouloughli, 1992]. ArabTAG est codée en XML (Extensible Markup Language). Elle couvre des structures

7. Laboratoire de Recherche en génie logIciel, Applications distribuées, systèmes Décisionnels et Imagerie intelligente

8. École nationale des sciences de l'informatique

elliptiques, d'autres anaphoriques et aussi des structures renfermant des subordonnées. Cette grammaire prend aussi en considération la variation des positions des éléments au sein des composants syntaxiques et le phénomène d'agglutination. Nous nous sommes intéressés à étudier les caractéristiques et la couverture d'ArabTAG. Cette étude constitue le point de départ de notre projet de thèse.

2.5.1 Caractéristiques d'ArabTAG

ArabTAG hérite de tous les fondements de base de TAG qui ont été accommodés aux spécificités de la langue arabe. Nous allons les présenter ci-dessous.

2.5.1.1 ArabTAG est générique

Seules les composantes de base des structures syntaxiques (sans prise en charge des compléments) sont représentées tout en prenant en considération l'ordre semi-fixe des composants dans ces structures. Les éléments essentiels de la phrase (ou du syntagme), présentés dans ArabTAG sont les suivants : Pour une phrase verbale c'est le verbe, le sujet et le(s) complément(s) d'objet tandis que pour une phrase nominale c'est le thème et le propos. Le but de cette généricité c'est de limiter le nombre des arbres élémentaires possibles.

2.5.1.2 ArabTAG est partiellement lexicalisée

ArabTAG contient un ensemble de 24 arbres élémentaires lexicalisés, réservés pour représenter les contextes possibles des mots outils jouant le rôle de "modificateurs" (particules de négation, modifieurs de temps (نواسخ)) portent des lexèmes de phrases et/ou de syntagmes. En outre, elle possède d'autres structures non lexicalisées, appelées aussi "modèles d'arbres". Ces derniers correspondent à des schèmes d'arbres élémentaires.
Le choix des arbres à lexicaliser est relatif au type de modification qu'apporte le modificateur à la composante syntaxique qu'il modifie. A titre d'exemples, les modificateurs du verbe لا, لن, لم (ne et pas) sont représentés par trois arbres élémentaires lexicalisés différents puisque chaque modificateur possède sa façon propre pour modifier le verbe. Cependant pour des prépositions telles que في (dans), إلى (à), على (sur), etc. elles sont représentées toutes avec deux arbres élémentaires non lexicalisés (dans la famille des syntagmes prépositionnels) étant donné que ces modificateurs ont tous le même effet sur les syntagmes nominaux auxquels ils s'attachent.

2.5.1.3 Richesse des structures de traits

Afin d'assurer une bonne gestion des compositions syntaxiques, à chaque nœud au sein d'un arbre élémentaire est associé sa propre structure de traits. Ces traits, dits d'unification, regroupent un ensemble de valeurs morphologiques, syntaxiques et compositionnelles qui décrivent les relations du composant syntaxique cible avec son contexte. Le tableau 2.9 englobe l'ensemble des traits d'unification de ArabTAG.

Trait	**Description**	**Valeurs possibles**	**Nœuds concernés**
Fonction	La fonction du nœud au sein de la structure élémentaire	Sujet, verbe, prédicat, COD, COI, Adjectif, qualifié, etc.	Tous les nœuds contenus dans les différentes structures élémentaires
Gen	Spécifie l'exigence ou l'impossibilité d'un accord en genre à l'intérieur de la structure ou lors de sa combinaison avec d'autres structures	+ : exigence d'accord en genre, - : impossibilité d'accord en genre	Tous les nœuds à catégorie nom, adjectif ou syntagme nominal
Nombr	Spécifie l'exigence ou l'impossibilité d'un accord en nombre à l'intérieur de la structure ou lors de sa combinaison avec d'autres structures	+ : exigence d'accord en nombre, - : impossibilité d'accord en nombre	Tous les nœuds à catégorie nom, adjectif ou syntagme nominal
Cas	Présente la voyelle de fin du mot	N : nominatif, A : accusatif, G : génitif	Tous les nœuds à catégorie nom, adjectif ou syntagme nominal
Pronom	Précise une exigence ou une interdiction d'un accord entre la forme (si c'est un verbe) avec son sujet (si nous sommes dans un contexte actif ou le suppléant de son sujet dans un contexte passif)	+ : exigence d'accord entre verbe et sujet (resp. suppléant du sujet), - : impossibilité d'accord entre verbe et sujet (resp. suppléant du sujet)	SV, V
Trans	La nécessité ou non de la présence d'un complément d'objet pour le verbe dans une structure de phrase verbale	I : Intransitif, TD : Transitif Direct, TI : Transitif Indirect, TD-TI : Transitif à la fois Direct et Indirect	SV, V
Voix	Présente la voix de conjugaison du verbe, si verbe il y a	Act : Active (مبني للمعلوم), Pas : Passive (مبني للمجهول)	PV, V
Proc/Enc	Présente la possibilité, l'impossibilité ou l'exigence de la présence d'un proclitique (respectivement enclitique) lié au nœud en question	+ : exigence de présence d'un proclitique (respectivement enclitique) - : impossibilité de présence d'un proclitique (respectivement enclitique), +/- : possibilité d'existence d'un proclitique (respectivement enclitique).	Pour le Proc : Tous les nœuds sans exception Pour l'Enc : Tous les nœuds sauf les nœuds racines PV et PN

TABLE 2.9: Organisation des traits de signification d'ArabTAG[Ben Fraj, 2010]

En plus de ces traits d'unification, des traits d'instanciation ont été définis. Ce sont des informations morphosyntaxiques nécessaires lors de la lexicalisation de la grammaire. Ces traits se résument en la valeur grammaticale (VG) et la forme du mot instanciant ainsi que les valeurs grammaticales et les formes des proclitique et enclitique s'ils existent.

2.5.2 Couverture d'ArabTAG

La première version d'ArabTAG comporte 241 arbres élémentaires non lexicalisées (modèles d'arbres) et 24 autres lexicalisés. Les modèles d'arbres sont répartis entre quatre grandes familles : phrases nominales, phrases verbales, syntagmes nominaux et syntagmes prépositionnels.

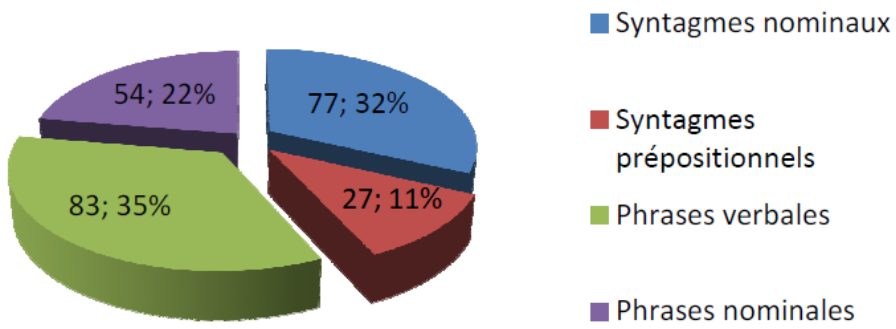


FIGURE 2.13: Structures présentées dans ArabTAG

Dans ArabTAG, toutes les structures des phrases nominales, respectant tous les agencements et variation des structures possibles, sont représentées comme par exemple Thème-Propos encore Propos-Thème. Ces deux composantes peuvent être des structures simples ou encore des structures plus complexes, par exemple des syntagmes nominaux, des syntagmes prépositionnels ou encore des phrases verbales. Cette famille de structures couvre 22% des structures non lexicalisées représentées en ArabTAG.
Pour ce qui est des phrases verbales, leur famille de structures comprend 35% des structures non lexicalisées représentées en ArabTAG. Ces structures couvrent uniquement les phrases à compositions minimale, à savoir : le verbe, le sujet et le(s) complément(s) d'objet direct ou indirect. Cependant, les différents agencements de ces composants sont représentés, par exemple : Verbe-Sujet-COD, Verbe-Sujet-COI, Verbe-COD-Sujet et Verbe-COI-Sujet. Cette famille inclut aussi les phrases à sujets elliptiques, les phrases à compléments d'objet directs enclitisés aux verbes, les phrases à verbes conjugués à la voix active et celles à verbes conjugués à la voix passive, les phrases verbales interrogatives etc.
ArabTAG tient compte de la diversité des structures syntagmatiques, spécialement nominales. On y trouve un ensemble assez complet de structures (32% des structures non lexicalisées représentées en ArabTAG) mettant en valeur les différentes sous-classes de syntagmes nominaux (voir figure 2.14) : syntagme simple ou syntagme à composition plus complexe.

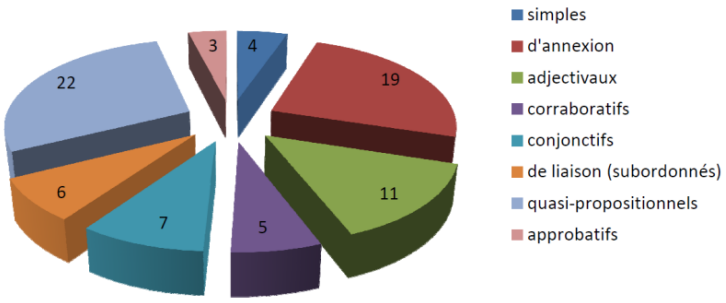


FIGURE 2.14: Répartition des syntagmes nominaux dans ArabTAG[Ben Fraj, 2010]

Ces sous-classes des syntagmes à composition couvrent les syntagmes suivants : le syntagme adjectival (مركب نعتي) (11 structures), le syntagme d'annexion (مركب إضافي) (19 structures), le syntagme quasi-propositionnel (مركب شبه إسنادي) (22 structures), le syntagme corroboratif (مركب توكيدي) (5 structures), le syntagme approbatif (مركب بدلي) (3 structures) le syntagme conjonctif (مركب العطف) (7 structures) et le syntagme subordonné (مركب موصولي) (6 structures). Les syntagmes prépositionnels, dont la famille de structures couvre 11% des structures non lexicalisées représentées en ArabTAG, présentent aussi une diversité. Ils englobent tous les syntagmes qui sont introduits par les mots outils : prépositions ou autres et qui possèdent une des deux structures générales (voir figure 2.15) :

— Composé d'un mot outil introduisant un syntagme nominal comme les exemples : في المنزلِ (dans la maison), إلا هو (sauf lui), etc.

— Un mot outil auquel a été lié un enclitique comme pour les exemples : به (avec lui), فيه (dans lui), etc.

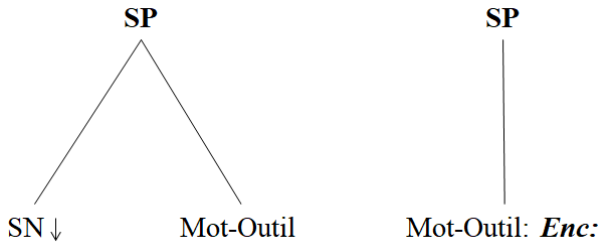


FIGURE 2.15: Structures générales d'un syntagme prépositionnel[Ben Fraj, 2010]

Précisons que dans ArabTAG, les syntagmes verbaux n'ont pas été illustrés. L'auteur [Ben Fraj, 2010] justifie son choix du fait, qu'en langue arabe, un verbe peut à lui seul constituer une phrase. Et donc, les syntagmes verbaux présentent eux-mêmes des phrases verbales.

2.5.3 Critiques d'ArabTAG

Nous avons étudié cette première version de la grammaire et nous avons relevé certaines limites pouvant se résumer comme suit :

— une couverture minimale : toutes les structures syntaxiques possibles ne sont pas décrites. Les structures enrichies avec des compléments (circonstanciel de temps, de lieu, etc) ne sont représentées. En effet, ces compléments sont facultatifs et peuvent prendre n'importe quelle position dans la phrase. Décrire ces composants dans le modèle d'origine, entrainera inévitablement l'augmentation du nombre d'arbres élémentaires et le risque de redondance.

— la représentation des formes agglutinantes dans les structures syntaxiques n'est pas bien prise en compte dans ArabTAG. Ces formes peuvent jouer des rôles dans la phrase et doivent être mises en relief afin d'améliorer la couverture du modèle grammatical développé.

— ArabTAG met l'accent sur les relations syntaxiques sans s'intéresser aux informations sémantiques, bien que la sémantique, tout comme la morphologie, possède une influence directe sur la syntaxe. En effet, l'interprétation syntaxique ne peut être complète que si l'on fait intervenir des informations sémantiques.

— ArabTAG n'est pas organisée en des structures factorisées hiérarchiquement. Elle est composée d'un ensemble d'arbres élémentaires sans qu'ils soient reliés entre eux. Dans le but de faciliter la maintenance et l'extension de la grammaire, il est primordial de structurer la grammaire en faisant intervenir divers phénomènes tels que l'héritage des structures ou la hiérarchie des patrons d'arbres.

2.6 Conclusion

La langue arabe est une langue importante en termes de diffusion et d'utilisation dans le monde. Elle présente des caractéristiques spécifiques qui compliquent la production des ressources numériques pour son traitement. En effet, sa riche morphologie associée à l'ordre semi-libre des mots et à l'omission des diacritiques (les voyelles) dans la plupart des textes arabes écrits affectent le processus d'analyse syntaxique et le rendent plus difficile. Malheureusement, comparés à d'autres langues telles que le français ou l'anglais, les outils et ressources génériques traitants de la langue arabe, telles que les grammaires, sont relativement rares et peu développés. Il est donc motivant de développer une grammaire qui reflète la richesse syntaxique et aussi sémantique de l'arabe.
L'objectif de cette thèse, rappelons-le, est de construire une grammaire d'arbres adjoints décrivant la syntaxe et la sémantique de la langue arabe. La construction d'une telle grammaire peut être abordée de différentes façons : l'extraction automatique d'une TAG à partir d'un corpus arboré ou la construction manuelle de la grammaire.
La méthode manuelle offre de meilleurs résultats que l'extraction à partir d'un corpus puisque la couverture de la grammaire construite est limitée à celle du corpus. Cependant, la construction manuelle est très coûteuse en termes de temps de mise en œuvre et de maintenance. De plus, il est difficile d'avoir une grammaire qui couvre toutes les structures syntaxiques d'un langage. À ce jour, il n'existe pas de grammaire à grande couverture de la langue arabe. Par conséquent, il est préférable d'adopter des approches réalisées de

façon automatique ou semi-automatique, en profitant des ressources déjà disponibles. Nous abordons ces approches dans le chapitre suivant.

Chapitre 3

Vers une production semi-automatique d'une grammaire d'arbres adjoints pour l'arabe

3.1 Introduction

Une grammaire TAG dans sa version lexicalisée LTAG [Joshi and Schabes, 1992] est constituée de milliers d'arbres ce qui entraine une redondance structurelle importante. De plus, le processus de son développement ainsi que sa maintenance sont très difficiles à réaliser puisque sa taille ne cesse d'augmenter considérablement. Par conséquent, il est préférable d'adopter des approches qui produisent une grammaire de façon automatique ou semi-automatique.
Certaines de ces approches, reposent sur le mécanisme d'extraction automatique de grammaires à partir de corpus arborés, par exemple les travaux de [Habash and Rambow, 2004]. Mais cette solution est contraignante puisque la qualité de couverture de la grammaire extraite est forcément limitée à celle du corpus. D'autres approches proposent de générer une TAG semi-automatique à partir d'une description réduite des règles de la grammaire cible. Cette description compacte de l'information grammaticale correspond à une méta-grammaire. Dans ce chapitre, nous nous intéressons à la production des grammaires d'arbres adjoints en utilisant une méta-grammaire.
Nous introduisons, dans un premier temps, la notion de production semi-automatique de grammaire LTAG et nous présentons brièvement quelques formalismes utilisés. Ensuite, nous consacrons la deuxième partie du chapitre à un formalisme en particulier à savoir XMG [Crabbé et al., 2013]. Nous détaillons, les fonctionnalités qu'il propose ainsi que ses implémentations.

3.2 Construction semi-automatique d'une grammaire LTAG

L'écriture d'une LTAG de taille importante est une tâche difficile et complexe. Plusieurs problèmes peuvent être rencontrés durant ce processus, parmi lesquels nous pouvons citer : le temps de développement, le risque d'absence de cohérence entre règles, les problèmes liés à l'évaluation de la grammaire et sa maintenance, etc.
En effet, nous rappelons que chaque arbre d'une LTAG comporte au moins un nœud feuille lexical (ancre). L'une des conséquences de cette lexicalisation est l'inévitable redondance des arbres de la grammaire [Vijay-Shanker and Schabes, 1992]. Une même structure syntaxique peut être employée plusieurs fois. De plus, certaines règles peuvent partager une sous-structure commune.
Pour illustrer ce problème, considérons l'exemple des deux arbres ci-dessous (figure 3.1). La partie encerclée représente la structure en commun de ces deux arbres. Cette structure redondante est un fragment d'arbre (sv : syntagme verbal) composé d'un sujet (syntagme nominale) suivi par son verbe.

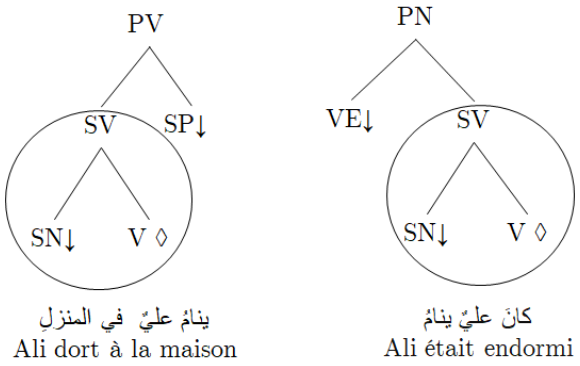


FIGURE 3.1: Exemple de réutilisation d'un fragment d'arbre

En plus de ces deux représentations syntaxiques, ce fragment d'arbre est utilisé dans chaque arbre de la grammaire comportant un syntagme verbal de la même structure, à savoir un sujet (SN) suivi par son verbe. Par conséquent la modification de telles structures entrainera la modification d'un grand nombre d'arbres (tous les arbres qui reprennent les mêmes sous-structures) et cela risque de causer des incohérences dans les règles de la grammaire.
Afin de faciliter la construction et la maintenance d'une LTAG tout en évitant ces inconvénients, diverses propositions de production semi-automatique de la grammaire ont été réalisées. Elles sont divisées en deux catégories : les approches basées sur les règles lexicales [Becker, 2000], [Prolo, 2002] et les approches basées sur des combinaisons de fragments (méta-grammaires) : [Candito, 1999], [Gaiffe et al., 2002]. [Xia, 2001], [Thomasset and De La Clergerie, 2005] et [Crabbé et al., 2013].
Becker [Becker, 2000] utilise un ensemble de règles lexicales (appelées méta-règles) pour produire automatiquement une grammaire TAG. L'inconvénient majeur de cette approche c'est le manque de contrôle sur la grammaire générée. En effet, il est difficile de vérifier

si la tâche de génération des nouveaux arbres TAG va s'achever, boucler sur une suite de règles, ou encore si la cohérence des règles est respectée.

Prolo [Prolo, 2002] a réussi à remédier à certains de ces problèmes en utilisant 21 métarègles pour générer 783 arbres correspondant aux 53 de ces familles de verbes. Cependant, parmi ces arbres générés il y'a ceux qui sont incorrectes alors que d'autres arbres n'ont pas été générés tout simplement.

Les approches méta-grammaticales ont été introduites pour la première fois par Marie Candito à la fin des années 1990 [Candito, 1996]. L'auteur a proposé un nouveau processus pour la génération semi-automatique d'une grammaire TAG à partir d'une description réduite. Cette dernière offre un haut niveau d'abstraction dans la description des structures syntaxiques de la grammaire. Elle capture les généralisations linguistiques apparaissant parmi les arbres de la grammaire et offre une décomposition fine en des blocs de construction syntaxiques, appelés aussi fragments élémentaires. Ces fragments, décrits sous forme de classes, sont combinés et structurés suivant une hiérarchie d'héritage en fonction des motivations linguistiques. Cette description réduite est appelée méta-grammaire.

Une méta-grammaire offre un moyen de partage important d'information afin d'éviter la redondance et les incohérences entre règles. De plus elle permet la généralisation de l'information contenue dans ces règles. La génération d'une grammaire à partir d'une méta-grammaire nécessite un compilateur méta-grammatical. Néanmoins, la forme de description méta-grammaticale proposée par Candito n'est pas flexible puisqu'elle est basée sur une hiérarchie tridimensionnelle de classes (les cadres de sous-catégorisation initiaux, les redistributions des fonctions syntaxiques et les réalisations syntaxiques de ces fonctions).

L'approche de [Gaiffe et al., 2002] introduit une plus grande flexibilité dans la description méta-grammaticale et propose un nouveau système de compilation de méta-grammaire basée sur un nombre arbitraire de dimensions (d'hiérarchies d'héritage de classes). Cependant, une des limites importantes de cette approche, qui est aussi observée chez celle de Candito, est le manque de contrôle des combinaisons de fragments d'arbres.

L'approche de [Xia, 2001] [1] propose un mécanisme de contrôle pour la combinaison de ses descriptions. A l'instar de l'approche de Candito, sa méta-grammaire est composée de trois constituants : un ensemble de cadres de sous-catégorisation canoniques, un ensemble de blocs élémentaires et un ensemble de règles lexicales de redistribution pour contrôler les combinaisons. Ces règles lexicales interviennent avant la combinaison des blocs afin de produire de nouveaux cadres de sous-catégorisation à partir des cadres canoniques définis.

[Thomasset and De La Clergerie, 2005] [2] propose une nouvelle approche de génération d'une TAG factorisée. [3] Pour produire de telles structures, le langage de définition de la méta-grammaire a été étendu par rapport aux approches précédentes. La description méta-grammaticale correspond à une hiérarchie de classes contenant des descriptions d'arbres étendues, des arguments éventuels et un ensemble de besoins et de ressources. Le système de contrôle est assuré par ces besoins et ces ressources. Lors de la compilation, les classes sont combinées en accumulant leurs contraintes, [4] jusqu'à l'obtention des classes neutres

1. Le système de UPenn de production semi-automatique de grammaires TAG.

2. MGCOMP le générateur d'arbres TAG factorisés.

3. Un arbre factorisé procède un nœud étiqueté par le symbole ## indiquant qu'il s'agit d'un entrelacement (l'ordre des fils de ce nœud est libre).

4. Les classes dont les contraintes sont accumulées doivent prendre en compte leurs conséquences logiques (par exemple précédence d'un nœud).

(annulation des besoins et des ressources). Les contraintes des classes neutres résultantes sont ensuite exploitées pour produire les arbres factorisés pour TAG. Toutefois, la vérification d'une telle grammaire avec un analyseur syntaxique est très compliquée. En effet, seul le système DyALog [Villemonte De la Clergerie, 2005] permet d'analyser ce type d'arbres factorisés.
La dernière approche, celle de [Crabbé et al., 2013] propose aussi un formalisme à base de combinaison de fragments d'arbres nommé XMG (eXtensible MetaGrammar). Il désigne à la fois un langage de description méta-grammatical, et le compilateur pour ce langage. XMG se distingue des approches antérieures par ses caractéristiques pertinentes (extensibilité et expressivité). De plus, il ne se limite pas à un formalisme syntaxique en particulier et génère aussi bien des TAG que les grammaires d'interaction (IG).

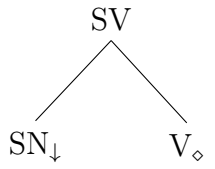
3.3 Formalisme méta-grammatical extensible XMG

Le formalisme XMG permet de générer toutes les structures associées aux mots du lexique à partir d'une description méta-grammaticale. Cette dernière est composée d'un ensemble d'abstractions représentant des fragments de la grammaire. Ces abstractions, ou encore descriptions de fragments, sont combinées au moyen d'un langage de contrôle des combinaisons. Elles sont aussi organisées dans une hiérarchie d'héritage pour favoriser la réutilisation de ces descriptions de manière flexible ainsi que la spécialisation de l'information.
En conséquence, il est possible de décrire finement les généralisations qui apparaissent dans une grammaire. Des plus, le degré élevé de factorisation donnée par XMG facilite le développement de la grammaire et sa maintenance puisqu'il suffira de mettre à jour et / ou ajouter des informations spécifiques à la description méta-grammaticale.

3.3.1 Définition de blocs élémentaires

Dans XMG, les fragments de la grammaire, appelés aussi blocs élémentaires, sont dénotés par les abstractions (nommées également classes). En effet, le concept d'abstraction permet d'associer un nom (nom de la classe) à un ensemble d'expressions (le contenu de la classe) de manière à pouvoir les réutiliser facilement. Ainsi, une description du fragment se fait au sein d'une classe comme suit :
(1) `Classe ::= Nom` $\rightarrow$ `Contenu`
(2) `Contenu ::= Description`
La description d'un fragment élémentaire est définie au moyen de contraintes de dominance et de précédence [Rogers and Vijay-Shanker, 1992] :
(3) `Description ::= x` $\lhd$ `y | x` $\lhd^+$ `z | y` $\lhd^*$ `z | x` $<$ `y | x` $<^+$ `z | y` $<^*$ `z | x=y | x[f:E] | x(p:E)`
Où x, y et z sont des variables de nœuds. $\lhd$,$\lhd^+$, $\lhd^*$ sont les prédicats binaires qui représentent respectivement la dominance immédiate, la dominance stricte et la dominance large et $<$, $<^+$, $<^*$ représentent respectivement la précédence immédiate, la précédence stricte et la précédence large. L'identification de nœuds se fait avec l'opérateur $=$.
x[f :E] permet d'associer un trait f de valeur l'expression E au nœud X. Quand à X(p :E), c'est l'association de la propriété p de valeur l'expression E au même nœud X.

Ces opérateurs servent à introduire les contraintes qui ont pour fonction d'assurer que les solutions générées soient des arbres bien formés.
Considérons l'exemple du fragment du syntagme verbal illustré dans la figure 3.1 :



Il est possible de définir une abstraction de ce fragment comme suit :
(4) Description ::= node SV [cat=sv]
∧ node SN [cat=sn]
∧ node V [cat=v]
∧ SV ◁ SN ∧ SV ◁ V ∧ SN < V
où SV, SN et V représentent des variables de nœuds. Les expressions "SV ◁ SN" et "SV ◁ V" indique que SV domine les deux nœuds SN et V. Tandis que "SN < V" signifie que SN doit précéder V. Les traits de chaque nœud sont définis entre deux crochets. Dans cet exemple, la catégorie de chaque nœud est précisée en associant une valeur au trait "cat". Soulignons que le symbole "∧" représente l'opérateur logique ET, permettant la combinaison des expressions définies.
En effet, XMG intègre un langage de contrôle des combinaisons de ces fragments basées sur les opérateurs logiques ET et OU.

3.3.2 Combinaison des blocs élémentaires

Les fragments ou abstractions définis, sont combinés au moyen des opérateurs logiques de conjonction et de disjonction.

3.3.2.1 Disjonction

L'opérateur de disjonction, dénoté par "∨", permet d'exprimer les alternatives entre les descriptions des fragments :
(5) Description ::= Desc ∨ Desc ∨ Desc ∨.....
Par exemple, il est possible de définir une abstraction sur la fonction syntaxique sujet en utilisant une combinaison disjonctive comme suit :
(6) Sujet → SujetCanon ∨ SujetRelatif
L'abstraction "Sujet" regroupe les réalisations possibles du sujet : le sujet sous forme canonique (SujetCanon) ou bien sous forme relative (SujetRelatif).

3.3.2.2 Conjonction

L'opérateur de conjonction "∧" permet l'accumulation des formules de descriptions :
(7) Description ::= Desc ∧ Desc ∧ Desc ∧.....
Considérons l'exemple suivant pour décrire les arbres des verbes transitifs qui illustrent la combinaison conjonctive d'un ensemble de fragments d'arbres :
(8) Transitif → MorphActif ∧ SujetCanon ∧ Objet

L'abstraction "Transitif" correspond à l'association de la description des fragments d'arbres de "MorphActif", de celle d'un sujet canonique "SujetCanon" et un objet décrit par une abstraction sur la fonction syntaxique objet "Objet".

3.3.3 Partage d'information entre classes

Une classe en XMG est une abstraction représentant des fragments de la grammaire. Elle est caractérisée par son identifiant (nom de la classe), d'éventuels paramètres ainsi qu'un ensemble de variables (locales, importées et /ou exportées).
Dans XMG, les identifiants ont par défaut une portée limitée à la classe dans laquelle ils sont déclarés. Mais cette portée peut être étendue grâce à la notion d'héritage. En effet, XMG s'inspire des techniques standard de programmation orientée objet et offre un moyen pratique de gérer la portée des identifiants en utilisant les déclarations d'importation / exportation.
De plus, le formalisme XMG utilise deux autres moyens de traitement des identifiants basé sur l'unification et la polarité (baptisé principe des couleurs dans XMG) [Muskens and Krahmer, 1998], [Duchier and Thater, 1999] et [Perrier, 2000].
Nous détaillerons cette variété de techniques de traitement des identifiants dans les sections suivantes.

3.3.3.1 Portée des variables

XMG offre un contrôle étendu de la portée des variables héritées. Il est possible de restreindre l'importation à des variables spécifiques et de renommer les variables importées. Ainsi, la classe fille peut réutiliser les variables, introduites localement dans sa classe mère, à condition qu'elles aient été exportées en utilisant le terme export symbolisé par "⇐". L'exportation des variables de la définition d'abstraction (1), se fait en associant une structure de traits contenant les noms des variables, à l'abstraction (1) :
`(9) Classe ::= <V1, . . . , Vn> ⇐ Nom → Contenu`
Cette définition traduit le fait d'associer à une classe un enregistrement contenant la liste des noms des variables exportées. Ces variables, nommées de V1 à Vn, deviennent donc accessibles en dehors de la classe elle-même.
Lors de l'héritage, l'import sélectif, dit import paramétré, permet d'assigner à un nom une portée semi-globale. La définition de l'import paramétré est réalisée comme suit :
`(10) Classe ::= <V1, . . . , Vn> ⇐ NomB ∠ NomA[I1, . . . , Im] → Contenu`
Dans cette définition, la classe correspond à l'association d'un nom NomB à un contenu. La classe hérite du contenu de la classe référencée NomA, mais en limitant les noms de variable importés de NomA à I1, . . . , Im.
XMG intègre aussi un système de renommage des variables à l'import :
`(11) Classe ::= <V1, . . . , Vn> ⇐ NomB ∠ NomA [I1 = J1, . . . , Im = Jm] → Contenu`
La classe NomB hérite du contenu de NomA. L'import des variables de cette dernière est restreint à I1...Im qui sont renommées respectivement J1...Jm. Ainsi les variables I1 . . . Im seront appelées en tant que J1...Jm dans la classe NomB.

3.3.3.2 Hiérarchie d'héritage

La notion de classe en XMG, offre la possibilité de gérer la redondance des fragments et obtenir une meilleure factorisation de ces blocs élémentaires. L'idée est de permettre à une classe B de réutiliser, dans sa description, le contenu d'une autre classe A en héritant de l'information spécifique dans A. Ainsi, certains nœuds peuvent être spécialisés en ajoutant de nouvelles fonctionnalités et/ou contraintes, etc.
Dans XMG, une classe peut en hériter d'une autre en utilisant le terme import symbolisé par "∠". Donc, si B hérite d'une classe A cela revient à dire que B importe A.
L'héritage est noté comme suit :
`(12) Classe ::= NomB ∠ NomA → Contenu`
La définition d'une classe correspond donc à l'association d'un nom NomB d'un contenu qui hérite du contenu associé à la classe désignée par NomA.
Considérons les deux fragments d'arbres de la figure 3.2 ci-dessous. Elle illustre respectivement un verbe et un verbe intransitif.

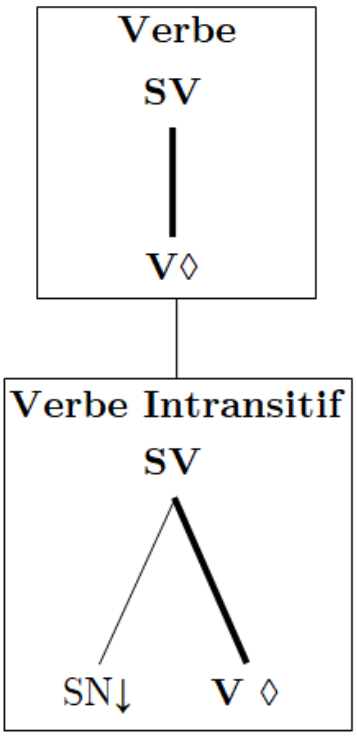


FIGURE 3.2: Exemple d'héritage d'un fragment d'arbre

Le fragment d'arbre représentant le verbe intransitif peut être considéré comme une spécialisation du fragment du verbe (structure en gras de la figure 3.2). Ainsi nous pouvons représenter la relation d'héritage comme suit :
`(13) Verbe Intransitif ∠ Verbe`
Le but de l'héritage est de rendre accessible directement le contenu d'autres classes. Ainsi, le contenu de la classe importée est mis à la disposition de la classe fille ce qui favorise l'extension de la portée de certaines variables.
XMG autorise aussi l'héritage multiple. Cette notion assure un plus haut degré de factorisation des blocs élémentaires puisqu'il rend possible l'utilisation de plusieurs blocs au

sein d'une même classe. Une classe peut donc hériter de plusieurs autres comme suit :
(14) Classe ::= Nom ∠ C1 ∧ C2 ∧ . . .Cn → Contenu
Les contenus des classes C1 ... Cn sont rendus accessibles à la classe fille de manière conjonctive. En d'autres termes, les descriptions définies dans les classes importées sont toutes incluses dans la description de la classe courante. Précisons qu'il est aussi possible, par héritage, de spécialiser une classe ou plusieurs classes en ajoutant de nouvelles fonctionnalités ou informations. En plus de l'héritage, XMG intègre un mécanisme de globalisation de la portée d'un nom appelé interface de classe.

3.3.3.3 Unification par interface

Le concept de l'interface est traditionnellement utilisé pour le partage des informations entre les dimensions (voir section 3.2.4 de ce chapitre) de la grammaire. Par exemple, l'interface permet d'établir un lien de correspondance entre la sémantique et la syntaxe. En XMG, la définition d'une interface au sein d'une classe correspond à une matrice de traits. Cette matrice permet d'associer un nom global (le trait) à une variable (la valeur du trait). Cette association est réalisée par l'opérateur "*=".
Considérons l'exemple suivant :
(15) NomA → . . . ?X . . . * = [g1 = ?X]
(16) NomB → . . . ?Y . . . * = [g1 = ?Y]
?X (resp ?Y) est une variable locale de la classe référencée NomA (resp NomB). Chaque variable de la classe est associée à l'identifiant global "g1". Ainsi, grâce à l'interface, la valeur associée à la variable locale ?X (resp ?Y) est maintenant accessible via l'identifiant global "g1".
Lorsque ces deux classes A et B sont conjointes ou encore lorsque l'une hérite de l'autre, leurs interfaces sont unifiées. En conséquence, si deux variables de chacune des deux classes ont le même nom global elles sont unifiées.
Par exemple, si les deux classes (15) et (16) sont conjointes, les deux variables ?X et ?Y seront unifiées puisqu'elles sont associées au même nom global "g1".

3.3.3.4 Le principe de couleur

Le formalisme XMG utilise un moyen très économique (pas de gestion d'espace de nom/ variables anonymes) pour automatiser la fusion des nœuds des fragments d'arbres. Il s'agit du mécanisme de polarisation de nœuds sous forme d'un langage de couleurs. Il est particulièrement utile lorsqu'une classe donnée doit être combinée avec de nombreuses autres classes. Ce mécanisme est aussi appelé "principe des couleurs".
Un principe en XMG, est utilisé pour calculer les structures grammaticales définies dans la méta-grammaire. Il correspond à un ensemble de contraintes qui s'appliquent aux modèles d'une description ce qui permet d'éviter la surgénération.
Le principe de couleur [Crabbé and Duchier, 2004] permet de guider la combinaison des fragments d'arbres définis, en attribuant une couleur black (noire), white (blanche) ou red (rouge) aux nœuds. Ces couleurs expriment les notions de besoin et de ressource :

— Un nœud noir est une ressource. Il est potentiellement saturé puisqu'il peut fusionner avec d'autres nœuds.

— Un nœud blanc est un besoin. Il devra impérativement fusionner avec un autre nœud ressource pour le compléter.

— Un nœud rouge est saturé. Il ne peut pas être fusionné avec un autre nœud.

Par conséquence, les combinaisons de nœuds autorisées sont limitées à l'ensemble illustré par la figure 3.3 :

	Noir	Rouge	Blanc	⊥
Noir	⊥	⊥	Noir	⊥
Rouge	⊥	⊥	⊥	⊥
Blanc	Noir	⊥	Blanc	⊥
⊥	⊥	⊥	⊥	⊥

FIGURE 3.3: Règles de combinaison des nœuds colorés

Un nœud coloré en noir peut être unifié avec 0, 1 ou plusieurs nœuds blancs et produit ainsi un nœud noir. Un nœud blanc doit être unifié avec un noir produisant un nœud noir. Enfin, un nœud rouge ne peut pas être fusionné avec un autre nœud. Ainsi, les arbres valides générés ont les nœuds colorés en noir ou en rouge.

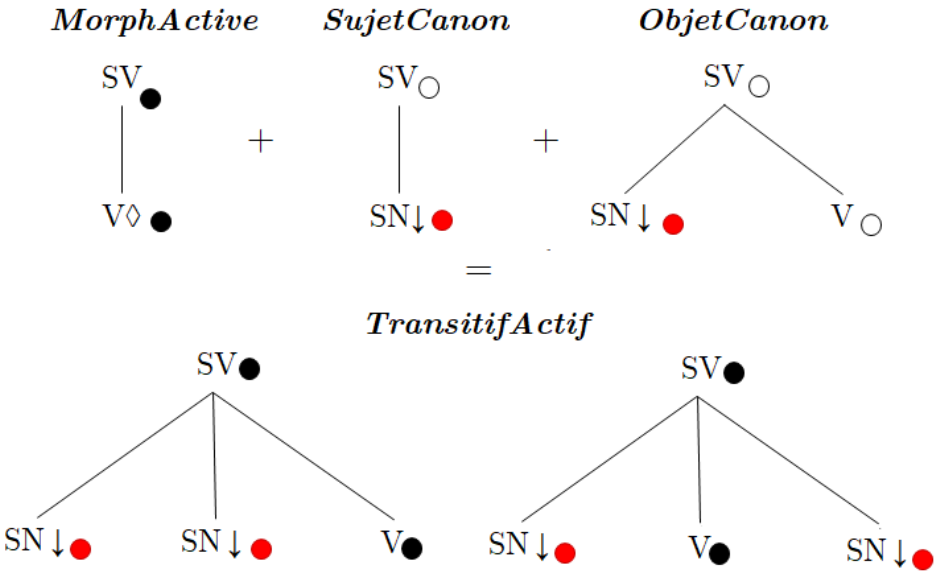


FIGURE 3.4: Exemple de combinaison des fragments d'arbres pour la classe TransitifActif

À titre d'exemple, considérons l'exemple de combinaison des fragments d'arbres illustré par la figure 3.4. Chaque nœud de ces fragments est coloré. Le fragment MorphActive contient uniquement des nœuds noirs. Ils sont utilisés pour saturer les nœuds blancs des

fragments de SujetCanon et de l'ObjetCanon. En d'autres termes le nœud blanc SV de SujetCanon s'attache au nœud noir SV de MorphActive. Tandis que, les deux nœuds blancs SV et V de ObjetCanon, fusionnent avec les nœuds noirs SV et V de MorphActive. Les deux modèles résultants [5] de TransitifActif ne contient que des nœuds rouges et noirs. Néanmoins, il est possible d'intégrer de nouveaux principes pour définir des contraintes dépendantes du langage. Ces contraintes auront pour but de produire des arbres qui respectent certains critères bien spécifiques du langage cible. Les traduire en principe(s) est une solution profitable pour une meilleure gestion des spécificités d'une langue, telle que l'arabe (voir chapitre 4).

3.3.4 Différents niveaux de description linguistique

Une caractéristique importante du système XMG est qu'il traite non seulement des structures arborescentes syntaxiques, mais également d'autres niveaux linguistiques de description. Imaginons que nous voulons étendre une méta-grammaire de manière à pouvoir décrire d'autres types d'informations que des arbres syntaxiques et produire par exemple une TAG augmentée de sémantique. Cela peut être fait en utilisant des dimensions qui distinguent différentes informations.
Chaque dimension est caractérisée par son propre langage de représentation et son propre processus de stockage. Néanmoins, toutes les dimensions ont le même processus de combinaison (et/ou).
Dans le système XMG, trois dimensions sont disponibles :

— **<syn> :** La dimension syntaxique pour décrire les arbres des fragments d'arbres des grammaires d'arbres adjoints et des grammaires d'interaction.

— **<sem> :** La dimension sémantique pour définir les structures prédicatives pour la sémantique plate [Bos, 1995].

— **<iface> :** La dimension interface qui permet le partage d'identifiants entre les différentes dimensions tel que l'interface sémantique-syntaxe (matrices d'attribut-valeur).

Le nombre de dimensions de XMG est limité à ces trois dimensions. Face à cette contrainte, on pourrait se demander sur la possibilité d'inclure des représentations particulières telles que les cadres sémantiques dans notre grammaire.
Ce manque de flexibilité a motivé le besoin d'apporter de la liberté dans la définition de nouvelles dimensions. C'est ainsi que XMG a été étendu en implémentant une méthode d'assemblage XMG-2 (voir section 3.3).

3.3.5 Définir une TAG avec XMG

En XMG, une méta-grammaire qui définit une TAG [Duchier et al., 2004] est décrite dans un fichier (ou un ensemble de fichiers) structuré en trois parties :

— **L'en-tête :** contient des éventuelles inclusions de fichiers, les déclarations de principes, de types et de traits.

5. L'ordre de précédence entre le verbe et sujet n'est pas spécifié.

— **Les descriptions des arbres élémentaires :** consiste à spécifier les classes définissant les fragments élémentaires. Une classe comporte trois parties : la déclaration de la classe, la gestion de l'espace des noms et la définition du contenu.

— **Les valuations :** la dernière étape d'une description méta-grammaticale qui consiste à exprimer la valuation des classes représentant les familles. Elle a pour effet de déclencher la combinaison des fragments définis (classes qui appellent des classes contenant des opérations de disjonction et/ou conjonction). Ainsi, pour chacune de ces classes, nous obtiendrons une description d'arbre cumulée pouvant conduire à la création de 0, 1 ou plusieurs arborescences TAG.

Nous présentons, dans la suite de cette section, la syntaxe concrète utilisée pour définir une classe de de fragments d'arbres en TAG.
Considérons la description méta-grammaticale des deux fragments de la figure 3.2 (section 3.2.3.2. de ce chapitre) :

```
(17) class Verbe[CouleurSV, CouleurV]
export  ?SV ?V
declare ?SV ?V
{
  <syn>
  {
    node ?SV (color=?CouleurSV) [cat=sv] {
      node ?V (color=?CouleurV, mark=anchor) [cat=v] }
  }
}

class VerbeIntransitif
import Verbe[black,black]
declare ?S ?X1
{
  <syn>
  {
    node ?S (color=red, mark=subst) [cat=sn, i=?X1];
    ?SV -> ?S;
    ?S >> ?V
  };
  <iface>{[arg0=?X1]}
}
value VerbeIntransitif
```

Une classe est définie avec le mot clé "class" suivi par son identifiant. Elle peut être paramétrée. Dans ce cas, ses paramètres sont entre crochets et séparés par des virgules. Dans cet exemple, la première classe est identifiée par son "Verbe" et possède deux paramètres CouleurSV, CouleurV. Tandis que la classe "VerbeIntransitif" n'est pas paramétrée.

```
class Verbe[CouleurSV, CouleurV]
class VerbeIntransitif
```

Dans "Verbe" les deux variables SV et V sont exportées comme suit : export ?SV ?V. [6] En d'autres termes, ces deux identifiants exportés sont désormais visibles par la classe "VerbeIntransitif". En effet, cette dernière hérite de la classe "Verbe". L'héritage est exprimé par l'instruction import et des valeurs sont attribuées aux variables passées en paramètre : import Verbe [black,black].
La classe "Verbe" (respectivement "VerbeIntransitif") déclare 2 identifiants : ?SV et ?V (respectivement ?S et ?X1 qui sont des variables locales).
Les identifiants, dans XMG, peuvent faire référence à un nœud (par exemple : SV, V et S), à une valeur d'une propriété d'un nœud (CouleurSV, CouleurV) ou à une valeur d'une fonctionnalité d'un nœud (?X1). Mais quel que soit l'identifiant auquel il fait référence, il doit avoir été déclaré auparavant, comme dans l'exemple (17) : declare ?SV ?V et declare ?S ?X1
Notons que le préfixe ? (pour les variables) et ! (pour les constantes anonymes) sont obligatoires lors de la déclaration.
Le corps de la classe, correspondant à la description de l'abstraction qu'elle réalise. Son contenu est délimité entre crochets et peut être composé de :

— une déclaration,

— une conjonction des déclarations représentées par "S1 ; S2",

— une disjonction des déclarations représentées par "S1 | S2".

La description syntaxique est définie suivant ce modèle <syn> formules. Les formules sont les descriptions des nœuds et leurs relations les uns avec les autres. Avec XMG, il est possible de donner un nom à un nœud en utilisant une variable, et lui associer des propriétés et / ou des fonctionnalités.
Une fois les nœuds du fragment d'arbre définis, leurs relations sont établies à l'aide des opérateurs suivants :

— -> domination immédiate

— ->+ dominance stricte

— ->* dominance large

— » précédence immédiate

— »+ précédence stricte

— »* précédence immédiate

— = équation de nœud

Concentrons-nous sur les descriptions syntaxiques de notre exemple :

```
<syn>
 {
   node ?SV (color=?CouleurSV) [cat=sv] {
     node ?V (color=?CouleurV, mark=anchor) [cat=v] }
}
```

6. Pour éviter le conflit de nom lors de l'exportation, l'identifiant peut être renommé. Cela se fait en utilisant export id1 = id1new.

Dans ce bout de code, la variable ?SV réfère à un nœud. [7] Ce nœud a une propriété : color (couleur du nœud) associée à la valeur de la variable ?CouleurSV. La structure d'entité [cat = sv] définie la catégorie du nœud SV. La variable ?V réfère au nœud qui est dominé par SV. V possède deux propriétés : celle de la couleur, associée à la valeur de la variable ?CouleurSV et mark qui indique que la valeur de la marque du nœud est ancre. Enfin la catégorie du nœud V est spécifiée par la structure [cat = v].

```
<syn>
{
  node ?S (color=red, mark=subst) [cat=sn, i=?X1];
  ?SV -> ?S;
  ?S >> ?V
}
```

Un seul nœud S est déclaré dont les propriétés indiquent que c'est un nœud de substitution (la valeur de la marque du nœud est substitution) et saturé (couleur rouge). Le trait "cat = sn" définit la catégorie du nœud et "i= ?X1" attribut la valeur de la variable ?X1 au trait i. Le nœud SV domine le nœud S (?SV -> ?S). Ce dernier précède V (?S » ?V). La dimension de l'interface est réalisée en suivant ce modèle <iface> formules. La déclaration <iface>[arg0= ?X1] de la classe "VerbeIntransitif" permet d'associer la variable X1 à la l'identifiant global "arg0". Chaque sous-formule définie peut être ajoutée de manière conjointe (en utilisant " ;") ou de manière disjonctive (en utilisant "|") à la description.

3.3.6 Processus de compilation de la méta-grammaire

XMG assure la production semi-automatique de la grammaire. Cette tâche est assurée par un compilateur qui a été implémenté dans XMG pour ce langage méta-grammaticale. Le processus de compilation est illustré par la figure 3.5 suivante.

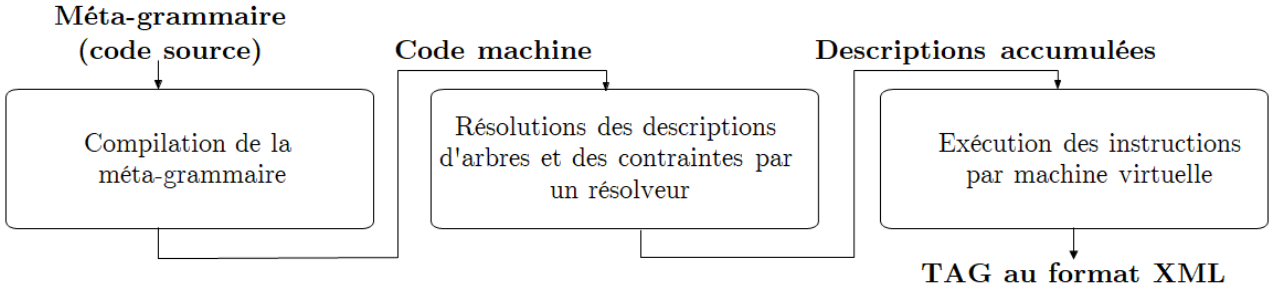


FIGURE 3.5: Processus de compilation de la méta-grammaire

Tout d'abord, le compilateur prend en entrée une description méta-grammaticale codée en XMG. Cette méta-grammaire est ensuite compilée en instructions (code machine) pour une machine virtuelle spécifique. Cette dernière est propre au système XMG. Elle a été inspirée de la machine abstraite de Warren [Aït-Kaci, 1991]. La machine virtuelle effectue des unifications entre les structures de données qui font référence aux descriptions des fragments (l'accumulation des nœuds, [8] structures des traits etc.). Le résultat produit

7. L'association d'une variable à un nœud est facultative
8. Les héritages de fragments et les combinaisons conjonctives et disjonctives sont développés.

à la fin de ce traitement consiste à un ensemble de descriptions d'arbres potentiellement totales. Ces arbres doivent être résolus afin de produire la grammaire attendue.
Cette étape est assurée par un résolveur [9] qui s'inspire des contraintes formulées dans [Duchier and Niehren, 2000]. Il a pour but de calculer les modèles de chacune des descriptions accumulées et de vérifier leur bonne formation (l'interprétation des contraintes tel que la résolution de descriptions colorées). Par exemple, concernant la dimension syntaxique, le résolveur de descriptions doit calculer l'ensemble des modèles d'arbres minimaux [10] correspondant à la description.
Finalement, la grammaire TAG, produite sous format XML, correspond à un ensemble de familles d'arbres élémentaires. Chaque arbre élémentaire possède un nœud ancré, qui sera remplacée lors de la lexicalisation par une forme fléchie.

3.4 Système XMG2 : définition modulaire de langages par assemblage de briques élémentaire

L'implémentation originale de XMG est rigide car elle permet une définition restreinte à trois dimensions de la description méta-grammaticale. Il est donc difficile d'inclure de nouvelles dimensions pour des représentations particulières telles que les cadres sémantiques. Afin de permettre la description d'un nombre illimité de dimensions, XMG doit incorporer un support pour ces nouvelles dimensions définies par l'utilisateur. Cela implique que le système doit être capable de définir formellement le langage de description pour cette dimension et d'interpréter les formules de ce langage pour produire des structures linguistiques valides.
Pour ce faire, XMG2 étend XMG en incluant un compilateur méta méta-grammaire (générateur de compilateur de méta-grammaire) [Petitjean, 2014] pour ces langages de description définis formellement. L'architecture modulaire de XMG2 permet donc aux contributeurs [11] de développer un ensemble de langages élémentaires appelés brique de langage. Une brique est la définition formelle d'un langage de description ainsi que la mise en œuvre de la procédure d'interprétation pour les formules (des arbres syntaxiques) de ce langage. Ces briques sont des unités réutilisables et assemblables. Ainsi, pour coder une nouvelle fonctionnalité du compilateur il suffit de créer une nouvelle brique sans prendre en considération son interface avec les briques existantes.
Le compilateur méta méta-grammaire peut alors compiler des ensembles de briques de langage, à la demande, qui peuvent ensuite être utilisés ensemble pour décrire différents niveaux de structures linguistiques. En d'autres termes, le compilateur méta-grammaire est maintenant compilé à partir de briques de langage (d'où le nombre arbitraire de dimensions).
Les 4 briques de langages actuellement disponibles dans XMG2 sont :

— B1 un langage de description de structures de traits,

— B2 un langage de description d'arbres à base de dominance/précédence entre nœuds,

9. Ce résolveur est implanté au moyen du paradigme de la Programmation par Contraintes (Constraint Programming).

10. Les modèles ou aucun nœud absent de la description n'est créé.

11. Le contributeur peut réaliser un assemblage de compilateur dédié à une tâche spécifique.

— B3 un langage de description de formules sémantiques plates,

— B4 un langage de description des cadres sémantiques [Lichte and Petitjean, 2015].

Nous nous intéressons à la dernière brique qui a permis de rajouter un nouveau type de représentation sémantique dans une méta-grammaire à savoir les cadres sémantiques.

3.4.1 Description des cadres sémantiques

La nouvelle brique (B4) pour la description des cadres sémantiques a permis le développement d'une nouvelle dimension <frame>. Cette dimension contient des descriptions de structures de traits typées. Ces structures utilisent des types conjonctifs. En d'autres termes, ces types ne sont pas atomiques, mais plutôt des ensembles de types élémentaires.[12] Lorsque deux structures de traits typées sont unifiées, le type de la structure résultante est déterminé par une hiérarchie de types. Si les deux types sont compatibles, selon cette hiérarchie, le type résultant sera l'union de ces deux types.
La hiérarchie de type est définie en deux étapes : tout d'abord les types atomiques sont déclarés en utilisant la syntaxe suivante : frame-types = t1,t2,...,tnou t1, t2, ..., tn sont des constantes.
Ensuite, ces types sont organisés selon une hiérarchie en spécifiant un ensemble de contraintes comme suit : frame-constraints = c1, c2,..., cn .ou c1, c2, ..., cn sont des contraintes de type. Il existe plusieurs types de déclaration de contrainte :

— Contraintes sur le sous-typage : t1 t2 ... tn -> tt1 tt2 ... ttn

— Contraintes d'incompatibilité : t1 t2 -> -

— Contraintes concernant les attributs t1 t2 ... tn -> c1 ... cn. c1 ... cn peuvent être des types suivants : contrainte d'existence (att :+), contrainte de valeur (att : val) ou égalité de chemin (att1 = att2).

A titre d'exemple, considérons la hiérarchie de type illustrée dans la figure 3.6 ci-dessous :

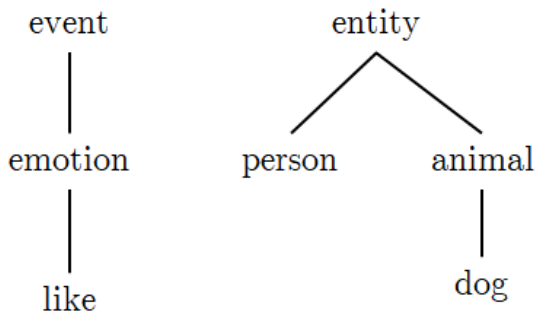


FIGURE 3.6: Exemple de hiérarchie de types

Tous les types sont déclarés comme suit :

```
frame-types = {event, emotion, like, entity, person, animal, dog}
```

12. La variable attribuée pour représenter une structure de traits est associée en plus de ses traits à un type. Ce type est également représenté par une variable attribuée.

Ensuite, nous définissons les contraintes de type en respectant la hiérarchie de la figure 3.6.

```
frame-constraints = {
emotion -> event,
entity event -> -,
like -> emotion,
person -> entity,
animal -> entity,
animal person -> -,
dog -> animal
}
```

"emotion" est un sous-type de "event" (emotion -> event) et "like" est un sous type de "emotion" (like -> emotion). L'incompatibilité entre le type "entity" et "event" est définie par la contrainte (entity event -> -). "Person" et "animal" sont des sous-types de "entity" (person -> entity, animal -> entity). Par contre, l'union de ces deux sous-types est interdite par la contrainte d'incompatibilité (animal person -> -). Enfin, "dog" est un sous type d'"animal" (dog -> animal).
Les cadres sémantiques sont décrits dans la dimension <frame>. Le type du cadre et les paires attribut-valeur (séparées par deux points) sont délimités entre crochets et séparés par des virgules. Les paires sont séparées par deux points. Par exemple, le cadre sémantique pour une personne peut être codé de la manière suivante :

```
  <frame>{
    ?X0[person]
  }
```

Avec ce type de définition, seule les types compatibles avec "person" pourront s'unifier avec cette structure. Nous pouvons affiner encore cette description en ajoutant une valeur spécifique au nom de la personne comme suit :

```
<frame>{
    ?X0[person,
      name: Ali]
  }
```

La valeur du prénom de cette personne est "Ali".
Considérons un autre exemple de description cadre sémantique pour décrire l'évènement "like" :

```
  <frame>{
    ?X0[like,
Actor: [person]
      Patient: [person]]
  }
```

Avec ce type de déclaration, nous avons imposé des contraintes de type sur les deux rôles "Actor" et "Patient". Ces deux rôles doivent être compatibles avec le type "person". En conséquence, si nous voulons tester l'unification de ce cadre sémantique avec d'autres structures de type "animal" ou "dog" l'unification échouera.

3.4.2 Processus de méta-compilation

Le schéma suivant (figure 3.7) illustre l'architecture de la chaîne de traitement modulaire de XMG2.

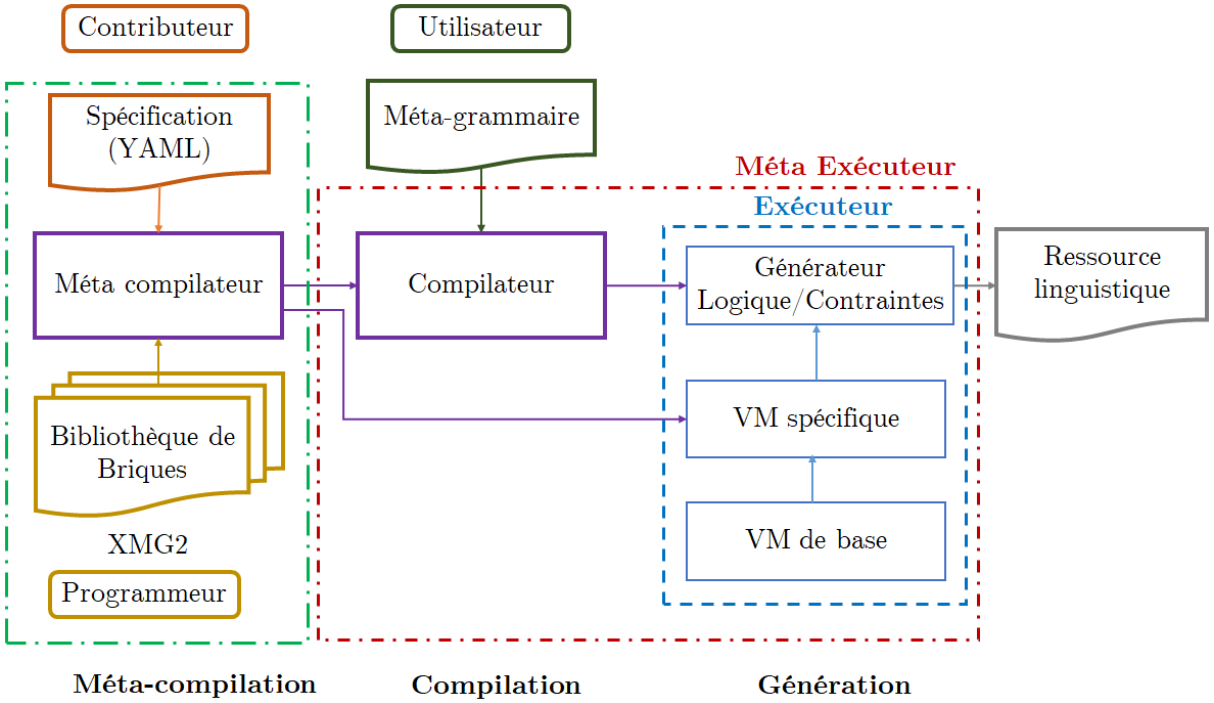


FIGURE 3.7: L'architecture modulaire de XMG2

Tout d'abord, un programmeur maitrisant le système XMG2, définit les nouvelles briques correspondantes aux nouvelles fonctionnalités.
À partir de la bibliothèque de briques de langage disponibles dans XMG2, les contributeurs (concepteurs de méta-grammaires) peuvent charger les briques nécessaires pour décrire leur formalisme cible. L'assemblage de ces briques se fait en déclarant dans un fichier YAML (liste de clés-valeurs), quelles briques doivent être chargées [Petitjean, 2014], [Petitjean et al., 2016].
Le méta-compilateur prend en entrée une définition formelle d'un langage de description, noté L, par assemblage de briques de langage. Ensuite, il génère un compilateur pour ce langage. Ce dernier prend en entrée une méta-grammaire écrite par l'utilisateur (ou linguiste) au moyen du langage L.
En fonction des principes utilisés dans cette méta-grammaire, celle-ci sera compilée par le compilateur (précédemment généré) pour produire un programme logique et des plugins de résolveur.
Chaque exécution du programme logique produit un ensemble de contraintes qui est soumis à un résolveur pour générer des solutions. Les solutions sont finalement traduites pour produire les éléments de la ressource linguistique.

3.5 Conclusion

Nous avons exploré diverses approches, visant à faciliter le développement et la maintenance des grammaires LTAG. Cette étude nous a permis de discerner l'importance des langages de description qui reposent sur les mécanismes d'abstraction. Parmi ces langages, nous nous sommes intéressés au formalisme XMG qui offre une génération semi-automatique des grammaires.
Ce formalisme a été utilisé pour développer plusieurs grammaires TAG électroniques pour le français [Crabbé, 2005], l'anglais et l'allemand. XMG se distingue avec ses caractéristiques qui sont particulièrement pertinentes pour la description des grammaires TAG :

— XMG est multi-formalisme, il ne se limite pas à un formalisme syntaxique en particulier et génère aussi bien des TAG que des grammaires d'interaction (IG).

— XMG est expressif, il permet de définir des descriptions factorisées de la grammaire. Ceci est très utile pour la description de différents phénomènes linguistiques tels que l'ordre des mots semi-libre dans la langue arabe.

— XMG est extensible, il peut être configuré pour définir un langage pour différentes dimensions. Ainsi il devient possible d'étendre la description méta-grammaticale pour différents niveaux de langage tel que la gestion simultanée des aspects syntaxiques et sémantiques du lexique. De plus, il permet de définir des modules de contraintes additionnelles pour la bonne formation des structures grammaticales produites.

La version étendue de XMG, XMG2, a apporté plus d'extensibilité en incluant un compilateur méta-grammatical. Ainsi il est devenu possible de définir un nombre illimité de nouvelles dimensions. Par exemple, la description de la morphologie et celle de la sémantique (en utilisant les cadres sémantiques).
XMG est donc particulièrement adapté pour décrire et générer, relativement rapidement, des grammaires d'arbres de couverture significative. Les fonctionnalités dont il dispose nous ont paru profitables pour la construction de notre TAG pour la langue arabe. Le processus de construction de cette grammaire en utilisant le formalisme XMG est détaillé dans le chapitre suivant.

Deuxième partie

Contribution

Chapitre 4

ArabTAG V2.0 : Une grammaire TAG avec dimension sémantique générée semi-automatiquement pour la langue arabe

4.1 Introduction

Ce travail de thèse vise à élaborer une grammaire formelle pour l'utiliser dans les applications du traitement automatique de la langue arabe. Contrairement aux autres grammaires qui traitent essentiellement de la syntaxe et sont, dans la plupart des cas, construites manuellement, nous proposons une description méta-grammaticale assez exhaustive permettant une représentation syntaxiquement et sémantiquement riche de cette langue.
En effet, au sein de cette méta-grammaire, une correspondance entre des composantes syntaxiques décrites de la phrase et des représentations sémantiques est réalisée. Ce lien entre la sémantique et la syntaxe est établi en utilisant une interface syntaxe-sémantique. Cette dernière permet de superviser la construction du sens de la phrase en unifiant les informations sémantiques de ses constituants.
Notre choix s'est porté sur des grammaires d'arbres adjoints. Ce formalisme a un pouvoir de représentation assez riche (les structures simples, complexes, combinatoires, partagées, etc.) et une capacité de traiter certains phénomènes linguistiques présents dans la langue arabe tels que les enchâssements. De plus, TAG facilite l'interfaçage entre le niveau syntaxique et le niveau sémantique. A titre d'exemple de définition d'interface syntaxe-sémantique au sein d'une TAG, nous pouvons citer les travaux de [Joshi and Vijay-Shanker, 2001], [Gardent and Kallmeyer, 2003], [Romero and Kallmeyer, 2005], [Parmentier, 2007] ou plus récemment [Kallmeyer and Osswald, 2013]. A notre connaissance, de tels travaux n'ont pas été menés sur l'arabe.
Ce présent chapitre est consacré à la présentation de notre approche. Il est organisé

comme suit : Dans un premier temps, nous nous intéressons à la description de la syntaxe de l'arabe au moyen de notre méta-grammaire. Ensuite, nous parcourons les phénomènes linguistiques de l'arabe couverts par la grammaire générée. Dans la deuxième partie de ce chapitre, nous introduisons et détaillons le processus d'intégration des informations sémantiques au sein de la méta-grammaire. Notre choix s'est porté sur la sémantique des cadres. Ce choix est motivé par la facilité de l'interfaçage entre le niveau syntaxique et le niveau sémantique.

4.2 Description de la syntaxe de l'arabe au moyen d'une méta-grammaire

La nouvelle version ArabTAG V2.0 [Ben Khelil et al., 2016], rappelons-le, est générée semi-automatiquement en utilisant le formalisme XMG. Tout d'abord, nous avons défini notre méta-grammaire manuellement en spécifiant les phénomènes linguistiques de l'arabe. Ensuite, cette description méta-grammaticale a été compilée automatiquement (avec le compilateur XMG) en grammaire TAG donnant lieu à ArabTAG V2.0.

4.2.1 Hiérarchies des fragment

Notre méta-grammaire, pour décrire la syntaxe de l'arabe, est répartie en trois parties : une description méta-grammaticale des phrases verbales, une description méta-grammaticale des phrases nominales et une description méta-grammaticale des syntagmes. Dans cette section, nous décrivons l'organisation hiérarchique de chacune de ces descriptions.

4.2.1.1 Phrases verbales

Nous nous sommes concentrés sur la modélisation des familles d'arbres des verbes (intransitif, transitif et ditransitif) en introduisant une organisation hiérarchique entre les fragments arborescents utilisés comme base de la représentation factorisée des phrases verbales dans notre grammaire. Cette organisation est schématisée dans la figure 4.1.
Le point de départ de cette organisation est l'abstraction EpineVerbe (C), illustrée dans la figure 4.2. Elle est paramétrée par une couleur C et contribue à un fragment d'arbre pour l'épine verbale.
Nous avons choisi de rajouter, au niveau de l'épine verbale, des points d'adjonction appropriés pour les adverbes qui peuvent être librement intercalés entre arguments. Nous reviendrons, en plus de détails, sur l'utilité de ce choix (voir section 4.1.2.2 de ce chapitre).

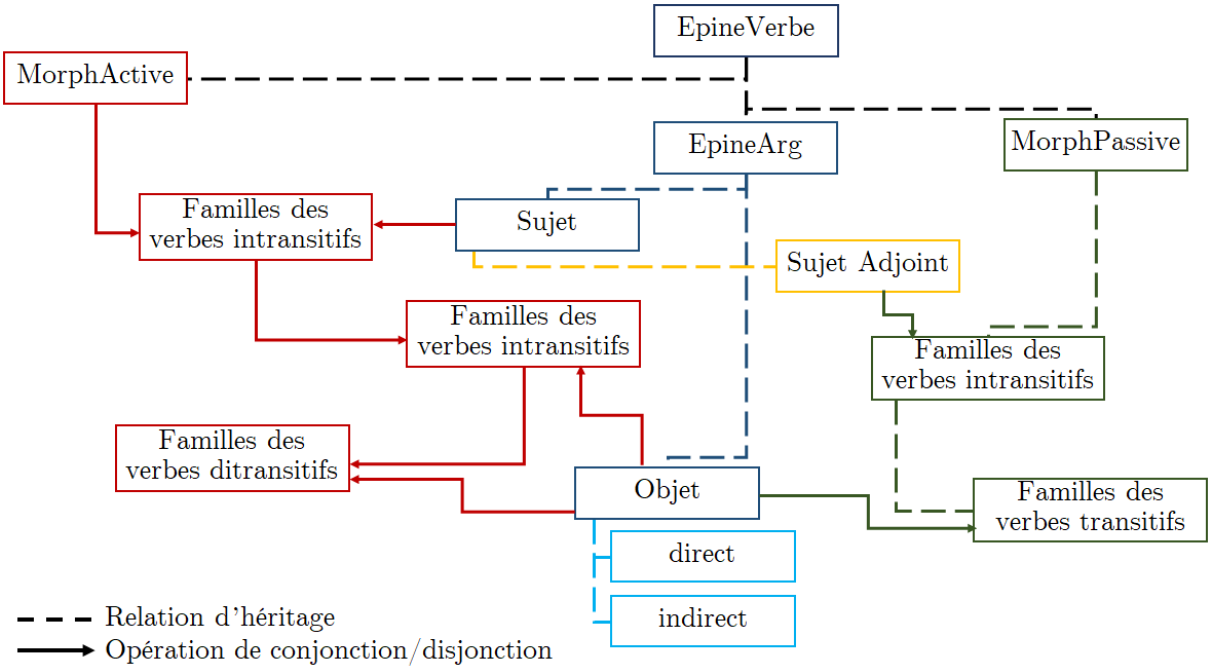


FIGURE 4.1: Organisation générale de la description méta-grammaticale des phrases verbales

$$\mathsf{EpineVerbe}(C) \longrightarrow \begin{array}{c} \mathrm{SV}^{C}\,\texttt{[cat=sv]} \\ | \\ \mathrm{AG}^{C}\,\texttt{[cat=advg]} \\ | \\ \mathrm{AD}^{C}\,\texttt{[cat=advd]} \\ | \\ \mathrm{V}^{C}_{\diamond}\,\texttt{[cat=v]} \end{array}$$

FIGURE 4.2: Description de la classe EpineVerbe

Deux formes d'instanciation d'EpineVerbe sont possibles : La classe MorphActive pour la forme active des verbes et la classe MorphPassive pour la forme passive :

$$\mathsf{MatrixClause} \longrightarrow \mathsf{EpineVerbe}(\mathrm{B})$$

$$\mathsf{MatrixClause} \longrightarrow \mathsf{EpineVerbe}(\mathrm{B}) \wedge \mathrm{V}\,\texttt{[voix=pas]}$$

La valeur de la couleur passée en paramètre est noire, permettant ainsi la combinaison de l'épine verbale avec d'autres fragments. Tandis que le trait voix (voix = pas) indique que la voix du verbe principal est un verbe passif.
Additionnellement, nous avons représenté deux autres formes de verbes avec clitiques illustrées dans la figure 4.3.

FIGURE 4.3: Fragments élémentaires pour les verbes avec clitiques

Le premier modèle, décrit les verbes précédés par une proclitique verbale. Quant au deuxième modèle, il représente le verbe auquel un enclitique est attaché (par exemple une anaphore représentant un objet du verbe).
Nous avons introduit EpineArg dans le but d'attacher l'épine verbale aux descriptions d'arbres de ses arguments :

$$\textsf{EpineArg} \longrightarrow [\mathrm{AG}] \Leftarrow \textsf{EpineVerbe}(\mathrm{W}) \wedge \mathrm{AD}^{\mathrm{R}}\texttt{[cat=advd]} \wedge \mathrm{AG} \triangleleft \mathrm{AD}$$

En effet, cette abstraction instancie EpineVerbe avec la couleur blanche, la forçant ainsi à s'unifier avec l'épine du verbe. [AG] ⇐ EpineVerbe (W) importe dans EpineArg la variable AG fournie par EpineVerbe et associe un nouveau nœud d'adjonction AD en vue de l'insertion facultative d'un adverbe après l'argument. Nous notons que AG ◁ AD spécifie que AG domine immédiatement AD, mais n'exige aucune contrainte de priorité.
Intéressons-nous, maintenant, aux arguments du verbe, à savoir le sujet et le(s) objet(s). Un sujet peut être sous forme elliptique, canonique ou relative.

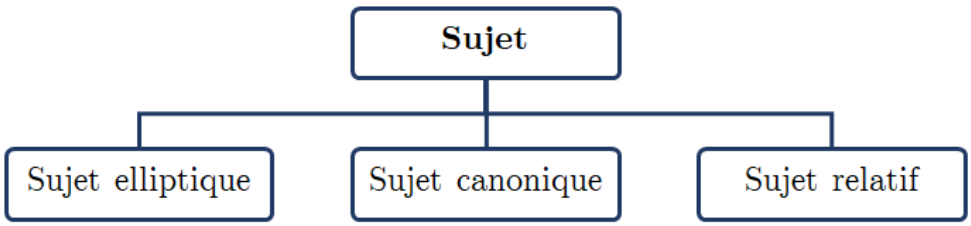


FIGURE 4.4: Les différentes classes du sujet dans une phrase verbale

Chacune de ces formes est modélisée par une classe spécifique.
Dans notre méta-grammaire, la classe du Sujet canonique représentant le sujet du verbe (cas nominatif) est décrite comme suit :

$$\textsf{SujetCanon} \longrightarrow [\mathrm{AD}] \Leftarrow \textsf{EpineArg}() \wedge \mathrm{SN}^{\mathrm{R}}_{\downarrow}\texttt{[cat=sn,cas=nom]} \wedge \mathrm{AD} \triangleleft \mathrm{SN}$$

SujetCanon instancie EpineArg et attache un nœud de substitution SN du sujet sous l'argument AD fourni par EpineArg.
Le sujet elliptique est défini de la manière suivante :

$$\textsf{Ellipse} \longrightarrow [\mathrm{V,SV}] \Leftarrow \textsf{EpineVerbe}(\mathrm{W}) \wedge \mathrm{SV}\texttt{[suj=s\_0]} \wedge \mathrm{V}\texttt{[voix=act]}$$

Enfin la classe SujetRelatif décrit deux modèles de syntagme subordonné pour la forme relative d'un sujet :

$$\textsf{SujetRelatif} \longrightarrow \mathrm{SN}^{\mathrm{R}}\texttt{[cat=sn, subcat=sn\_sub]} \;\vee\; \mathrm{SN}^{\mathrm{R}}\texttt{[cat=sn, subcat=sn\_sub]}$$
$$\mathrm{S}^{\mathrm{R}}_{\downarrow}\texttt{[cat=subor]}\ \mathrm{SV}^{\mathrm{W}}_{\downarrow}\texttt{[cat=sv]} \qquad \mathrm{S}^{\mathrm{R}}_{\downarrow}\texttt{[cat=subor\_p]}\mathrm{SV}^{\mathrm{W}}_{\downarrow}\texttt{[cat=sv]}$$

De même que le sujet, un objet peut être réalisé sous différentes formes : un syntagme nominal, un syntagme prépositionnel, un syntagme verbal, syntagme subordonnée (objet relatif) ou une enclitique.

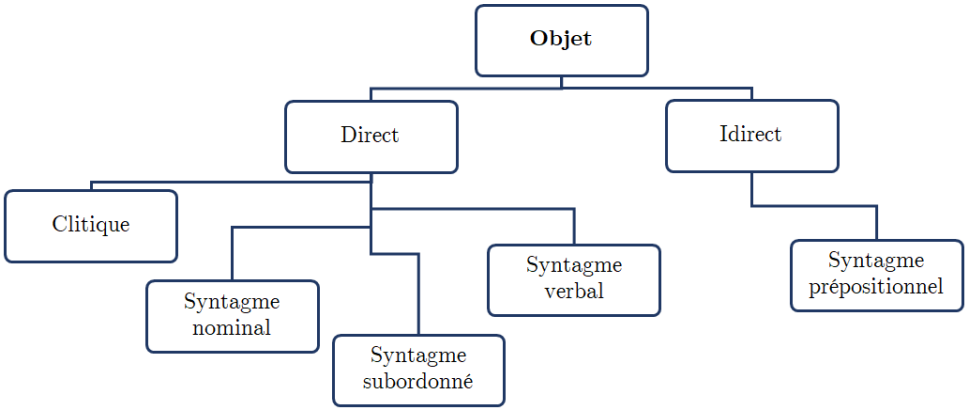


FIGURE 4.5: Hiérarchie des différentes classes de l'objet pour une phrase verbale

Chacune de ces variantes est représentée par une classe spécifique.
Le complément d'objet direct est exprimé par un syntagme nominal [cat=sn] dont le cas est accusatif [cas=acc].

$$\textsf{ObjetCanonSN} \longrightarrow [\text{AD, V}] \Leftarrow \textsf{EpineArg}() \wedge \text{SN}^{\text{R}}_{\downarrow}\texttt{[cat=sn, cas=acc]} \wedge \text{AD} \lhd \text{SN} \wedge ((\text{V}\texttt{[oclit=}-\texttt{]} \wedge \text{V} \prec^{+} \text{SN}) \vee (\text{V}\texttt{[oclit=}+\texttt{]} \wedge \text{SN} \prec^{+} \text{V}))$$

Ce complément d'objet peut se placer avant le verbe. Dans ce cas, une anaphore (référant au complément d'objet) doit s'attacher à la fin du verbe. Cette contrainte est exprimée avec le trait [oclit=+] qui indique la présence d'un enclitique liée au nœud en question.
La classe pour un complément d'objet sous forme d'un syntagme verbale est décrite comme suit :

$$\textsf{ObjetSV} \longrightarrow [\text{AD, V}] \Leftarrow \textsf{EpineArg}() \wedge \text{SN}^{\text{R}}_{\downarrow}\texttt{[cat=sv]} \wedge \text{AD} \lhd \text{SN} \wedge \text{V} \prec^{+} \text{SN}$$

Tandis que la classe pour un complément d'objet relatif est la suivante :

$$\textsf{ObjetRelatif} \longrightarrow \overbrace{\text{SN}^{\text{R}}\texttt{[cat=sn, subcat=sn\_sub]}}^{\text{S}^{\text{R}}_{\downarrow}\texttt{[cat=subor]}\ \text{SV}^{\text{W}}_{\downarrow}\texttt{[cat=sv]}}$$

La classe suivante indique que le complément d'objet peut être un enclitique au verbe :

$$\textsf{ObjetCanonClit} \longrightarrow [\text{V}] \Leftarrow \textsf{EpineVerbe}(\text{w}) \wedge \text{V}\texttt{[oclit=+]}$$

Enfin, le complément d'objet indirect est exprimé par un syntagme prépositionnel SP :

$$\textsf{ObjetIndCanon} \longrightarrow [\text{AD, V}] \Leftarrow \textsf{EpineArg}() \wedge \overbrace{\text{SP}^{\text{B}}\texttt{[cat=sp]}}^{\text{P}^{\text{R}}_{\diamond}\ \text{SN}^{\text{R}}_{\downarrow}\texttt{[cat=sn, cas=gen]}} \wedge \text{AD} \lhd \text{SP}$$

Nous avons utilisé la transitivité du verbe comme un critère fondamental pour l'héritage et nous avons combiné ces fragments afin d'obtenir les trois familles de verbes (intransitif, transitif, ditransitif). Chacune de ces classes capture les différentes réalisations syntaxiques possibles entre les différentes structures de la phrase. Elles sont organisées comme suit :

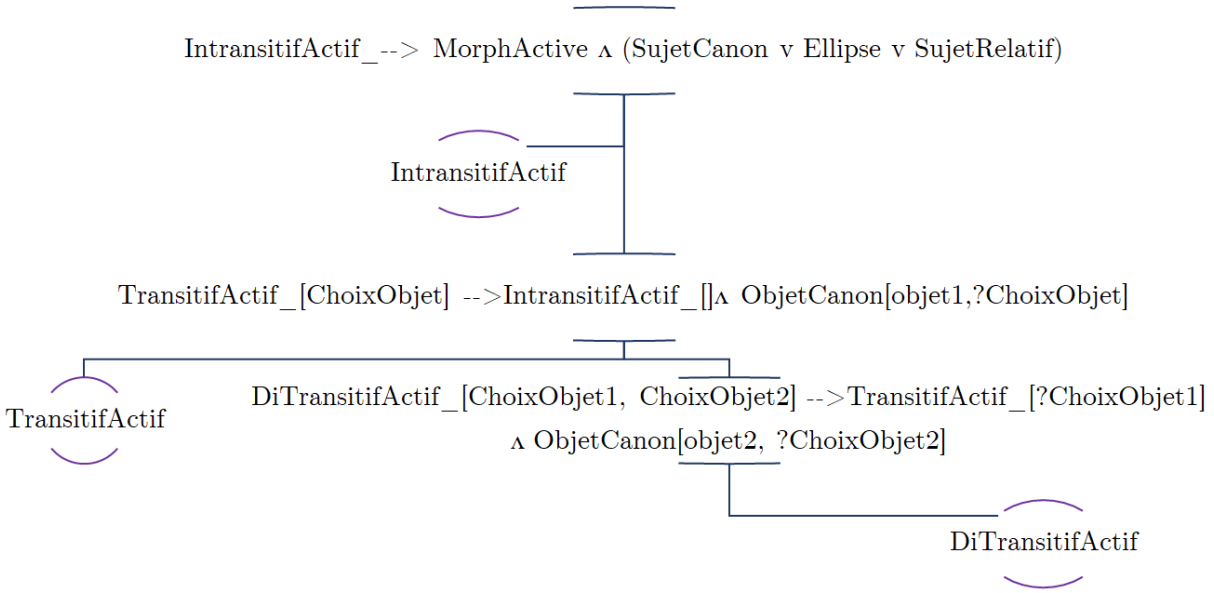


FIGURE 4.6: Hiérarchie d'héritage des familles des classes verbales sous forme active

La classe Morphactive définit le fragment d'arbre élémentaire qui constitue l'épine verbale de la phrase. Suite à une opération de conjonction entre cette classe et les classes du sujet (SujetCanon, sujet relatif ou Ellipse du sujet), nous obtenons la famille Intransitive. La famille Transitive est obtenue en combinant la classe Intransitive et la classe Objet. Cette dernière représente une disjonction entre le complément d'objet direct (sous forme canonique, relative ou encore clitique) et indirect.
Finalement, nous obtenons la famille DiTransitive en combinant Transitive avec les compléments d'objets seconds.

4.2.1.2 Phrases nominales

La deuxième partie de la description méta-grammaticale définit les fragments d'arbres propres aux phrases nominales. L'organisation de ces fragments est illustrée par la figure 4.7.
De la même manière que la phrase verbale,[1] nous avons défini une abstraction EpinePN (paramétrée par une couleur C) qui contribue à un fragment d'arbre pour l'épine nominale (Thème). Cependant, cette dernière est instanciée par deux différentes classes EpineTheme et EpineThemeV. En effet, nous avons fait le choix de distinguer ces deux classes afin de permettre la représentation des phrases nominales modifiées (les phrases qui commencent par un verbe d'existence ou un verbe de certitude).
"Theme" regroupe l'ensemble de réalisations possibles des thèmes pour les phrases nominales simples et modifiées. De ce fait, cette classe est paramétrée par une variable du cas permettant ainsi de définir le thème nominatif ou accusatif ainsi que l'épine verbale qui correspond au thème.
"Propos" possède trois paramètres : le cas du propos, le cas du thème et l'épine du verbe. Cette classe regroupe l'ensemble des réalisations possibles du propos : adjectif, syntagme

1. Contenant des points d'adjonction appropriés pour les adverbes.

nominal, syntagme prépositionnel, complément circonstanciel etc.

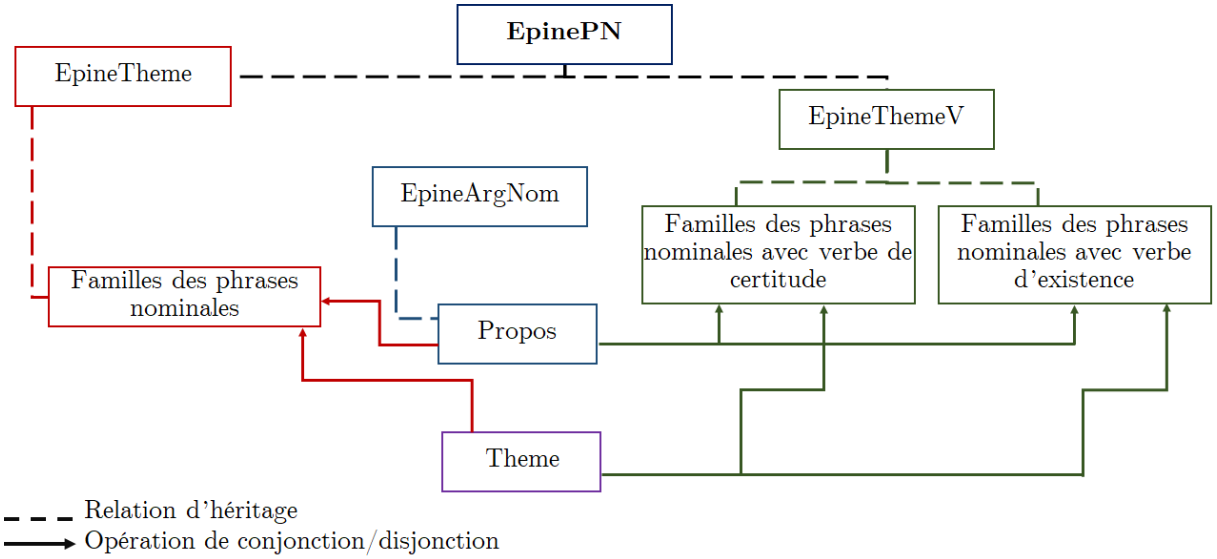


FIGURE 4.7: Organisation générale de la description méta-grammaticale des phrases nominales

La famille des phrases nominales correspond donc à la combinaison de l'épine verbale EpineTheme, le Thème nominatif et le propos nominatif.

PhraseNominale $\longrightarrow$ ET = EpineTheme[black, nom] ; Theme [nom, ?ET] ;
Propos[nom,nom, ?ET]

Tandis que les phrases nominales modifiées sont distinguées en deux familles comme suit :

PhraseNominaleVE $\longrightarrow$?ET = EpineThemeV[black, v_e, nom] ; Theme [nom, ?ET] ;
Propos[acc,nom, ?ET]
PhraseNominaleVC $\longrightarrow$?ET = EpineThemeV[black, v_c, nom] ; Theme [nom, ?ET]

PhraseNominaleVE correspond à la famille des phrases nominales modifiées par un verbe d'existence. L'épine du thème des phrases modifiées est combinée avec la classe Theme (nominatif) et la classe Propos (accusatif).
Quant à PhraseNominaleVC, la famille des phrases nominales modifiées par un verbe certitude, est définie par la conjonction de l'épine du thème des phrases modifiées, la classe Theme (accusatif) et la classe Propos (nominatif).

4.2.1.3 Syntagmes

La méta-grammaire définit les différents types de syntagmes que nous avons cités précédemment (voir 2.1.2.1.) à savoir les syntagmes nominaux (le syntagme d'annexion, le syntagme adjectival, le syntagme corroboratif, le syntagme approbatif, le syntagme de mode, le syntagme conjonctif et le syntagme quasi-propositionnel), les syntagmes prépositionnels et les syntagmes subordonnées. Chaque catégorie de syntagmes est décrite par au moins un fragment élémentaire.

Nous nous sommes aussi assurés d'intégrer des descriptions pour les structures élémentaires de la phrase tel que les noms propres, noms communs (déterminé ou non) pronoms, adjectifs, etc.

4.2.2 Phénomènes syntaxiques traités par ArabTAG V2.0

Dans cette section, nous nous intéressons aux phénomènes linguistiques spécifiques à la langue arabe traités dans ArabTAG V2.0 [Ben Khelil et al., 2018b] tels que la variation des positions des éléments au sein des composants syntaxiques, les compléments optionnels de la phrase, les règles d'accord, les formes agglutinées ainsi que les structures complexes (enchâssées et croisées).

4.2.2.1 Ordre semi-libre des mots

Une grammaire TAG est adéquate pour le traitement automatique d'une langue naturelle à ordre semi-fixe telle que l'arabe.

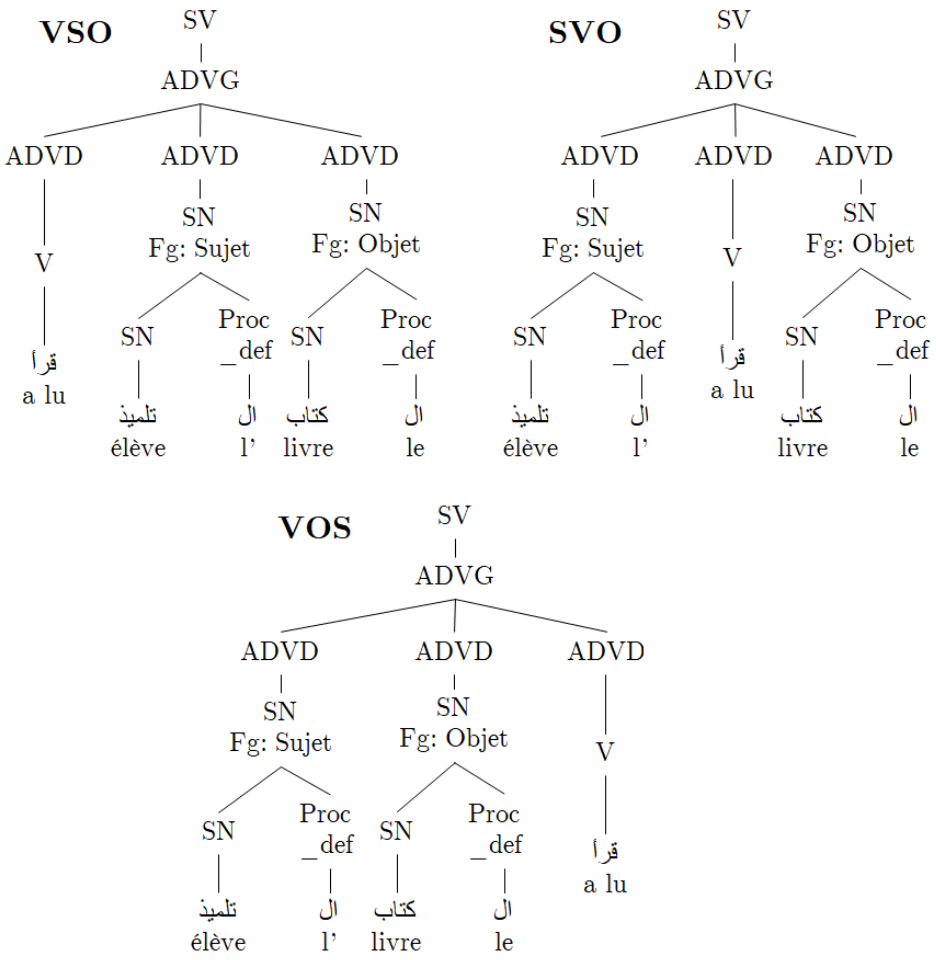


FIGURE 4.8: Ordre de mot libre de la phrase (L'élève a lu le livre)

Grâce à ses opérations d'adjonction et/ou de substitution, elle peut combiner les structures arborescentes sans prendre en considération l'ordre des combinaisons. Ces deux opérations peuvent être exécutées dans un ordre libre et ainsi former des phrases à multiples structures syntaxiques.
Par ailleurs, en utilisant les fonctionnalités de XMG, nous avons réussi à gérer ce phénomène au sein de notre description méta-grammaticale. Pour ce faire, nous avons évité d'imposer des contraintes de priorité entre les nœuds dont le changement d'ordre n'affecte pas la cohérence du sens de la phrase. Ainsi, lors des conjonctions et/ ou disjonction des fragments d'arbres définis, toutes les combinaisons générées couvriront toutes les structures arborescentes correspondantes à tous les emplacements possibles des nœuds sans contrainte de précédence.
Notre grammaire fournit tous les modèles d'arbre de ces changements d'ordre. A titre d'exemple, considérons la phrase suivante : "قرأ التلميذ الكتاب" (L'élève a lu le livre) dont la structure est VSO. ArabTAG V2.0 accepte les deux autres modèles résultat des permutations entre le verbe, le sujet et objet : "التلميذ قرأ الكتاب" (SVO) et "قرأ الكتاب التلميذ" (VOS). Ces modèles d'arbre sont présentés dans la figure 4.8.
Cependant, lors de la vérification des modèles arborescents qui traitent de l'ordre des mots, nous avons relevé un problème de surgénération des modèles des phrases verbales avec deux compléments d'objets (verbe ditransitif). Ce problème est illustré dans la figure 4.9.

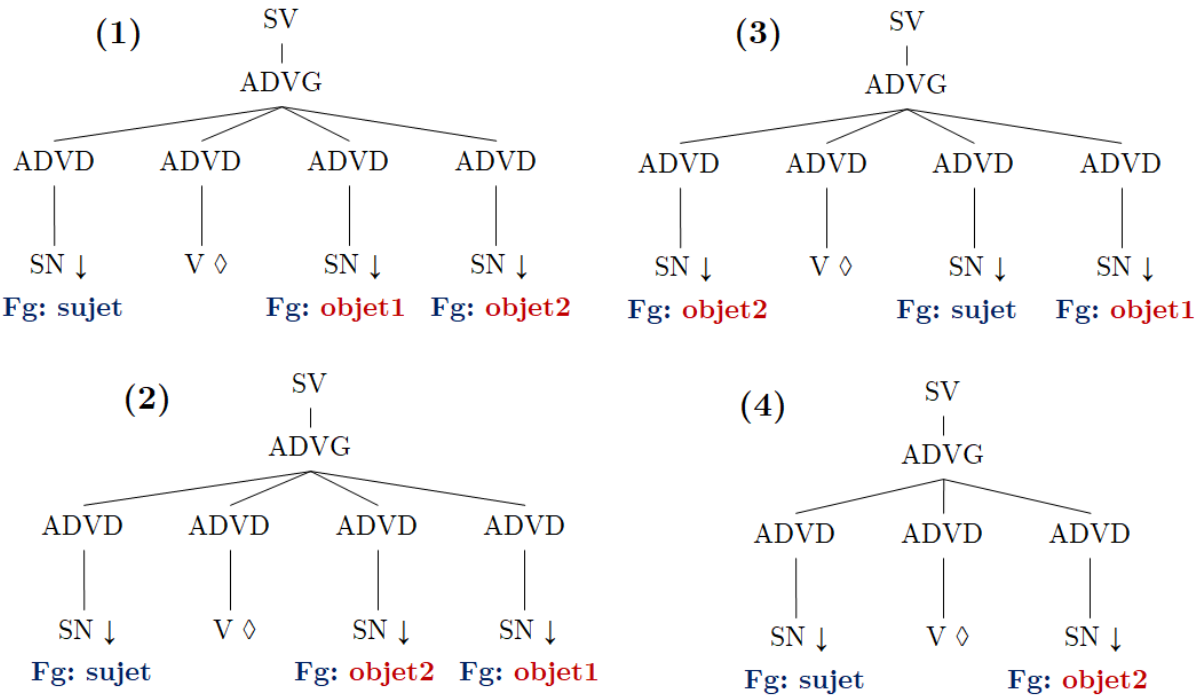


FIGURE 4.9: Exemple de redondance d'une structure arborescente de la grammaire

En effet, nous remarquons que les trois premiers modèles des arbres de la figure 4.9 ont la même structure. La seule différence est l'ordre des deux compléments d'objets (de même catégorie). Cependant, le premier complément d'objet doit toujours précéder le

complément d'objet second. Afin de prévenir la génération de ces modèles redondants et respecter cet ordre de précédence, nous avons défini et intégré deux nouveaux principes dans XMG : "precedes" et "requires".
Le premier principe "precedes" permet de gérer l'ordre de précédence entre deux nœuds X et Y. X possède une propriété p1 et Y possède une propriété p2. Ce principe est déclaré de la manière suivante :

```
use precedes with (p1=v1, p2=v2) dims (d).
```

Cette déclaration signifie que si un nœud X possède la propriété p1=v1 et un nœud Y avec p2=v2 alors X doit obligatoirement précéder Y. Tous les modèles générés par le compilateur seront conformes à cet ordre de précédence établi par le principe.
Toutefois, ce principe ne contraint pas la génération du modèle arborescent (4) de la figure 4.10. Cet arbre correspond à la structure d'une phrase avec un verbe transitif possédant un complément d'objet. Or ici il s'agit du complément d'objet second. Nous avons donc mis en place un deuxième principe "requires". Ce principe exige impérativement la présence d'un nœud X pour que le nœud Y existe.
Ce principe est déclaré comme suit :

```
use requires with (p2=v2, p1=v1) dims (d)
```

Si un nœud Y dont la propriété est "p2=v2" existe alors un autre nœud X ayant "p1=v1" doit exister.
En combinant ces deux principes comme suit :

```
use precedes with (fg=objet1, fg=objet2) dims (syn)
use requires with (fg=objet2, fg=objet1) dims (syn)
```

Les combinaisons des fragments élémentaires sont contrôlées. À la fin de la compilation, les modèles d'arbres syntaxiques sont ceux qui remplissent les deux conditions, c'est à dire : le complément d'objet (objet1) précède toujours le complément d'objet second (objet 2) et ce dernier (object 2) existe seulement si la présence du premier complément d'objet est vérifiée (objet 1). Ainsi, si nous revenons aux deux modèles d'arbres de la figure 4.10 seul le premier modèle (1) est généré comme l l'illustre la figure 4.10.

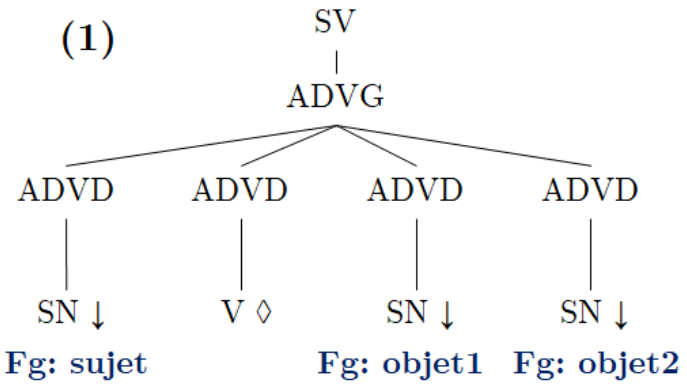


FIGURE 4.10: Le modèle gardé suite à l'application des principes "precedes" et "requires"

4.2.2.2 Adverbes et compléments optionnels

L'adjonction dans TAG permet l'insertion d'une structure complète au niveau d'un nœud intérieur d'une autre structure complète. Ce procédé est un moyen de gestion très naturel des adverbes et des compléments optionnels en langage naturel. En arabe, les adverbes et les compléments optionnels (les compléments circonstanciels du temps, les compléments circonstanciels du lieu, complément de mode, cause, etc.) peuvent être librement placés entre les constituants de la phrase. Nous avons décidé de fournir, à ces compléments, deux points d'adjonction appropriés dans notre description méta-grammaticale : ADVG (adverbe gauche cat= advg) et ADVD (adverbe droit cat= advd).

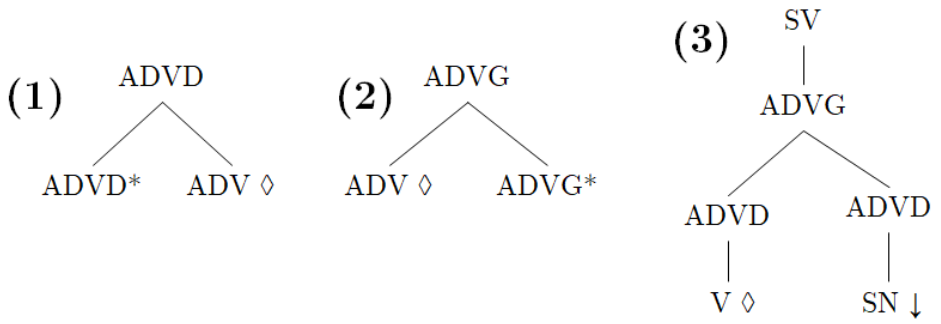


FIGURE 4.11: Ensemble de fragments d'arbres pour la gestion des adverbes

Le nœud ADVG du modèle (3) de la figure 4.11 est un point d'adjonction permettant l'insertion d'un adverbe (conforme au modèle (1)), ou un complément optionnel au début de la clause tandis que ADVD permet d'insérer un adverbe, du modèle (2), (ou un complément optionnel) après un verbe ou un argument.

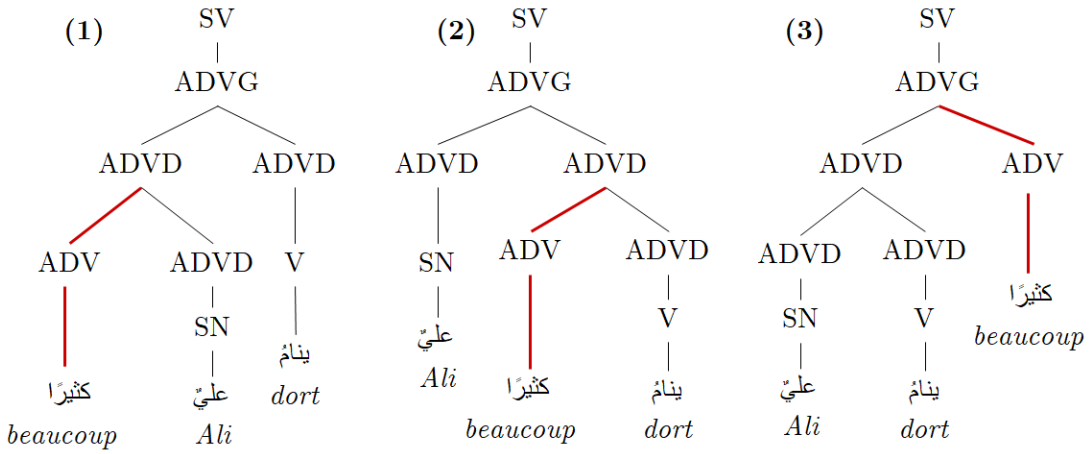


FIGURE 4.12: Insertion de l'adverbe (Beaucoup) dans la phrase (Ali dort)

À titre d'exemple, nous pouvons ajouter l'adverbe "كثيرًا" (beaucoup) dans la phrase "ينامُ عليٌ" (Ali dort) avant le sujet "ينامُ كثيرًا عليٌ" (dort beaucoup Ali) ou après le sujet

"ينامُ عليٌ كثيرًا" (dort Ali beaucoup). L'insertion est réalisée au niveau du noeud d'adjonction ADVD de l'adverbe droit. De même, cet adverbe peut se placer en début de la phrase. Grâce au noeud d'adjonction ADVG de l'adverbe gauche, "كثيرًا" (beaucoup) est inséré avant le verbe "ينامُ" (dort). Nous obtenons ainsi la phrase "كثيرًا ينامُ عليٌ" (beaucoup dort Ali). Les arbres syntaxiques d'ArabTAG V2.0 correspondants à ces trois phrases sont illustrés dans la figure 4.12.

4.2.2.3 Formes agglutinées

TAG permet la gestion du phénomène d'agglutination grâce à son pouvoir de contrôle. En effet, les structures de traits, définies au sein de ses arbres élémentaires, peuvent contenir des informations morphologiques et syntaxiques. Ces traits permettent d'assister la procédure d'analyse syntaxique et ainsi lever les ambiguïtés pouvant y survenir.

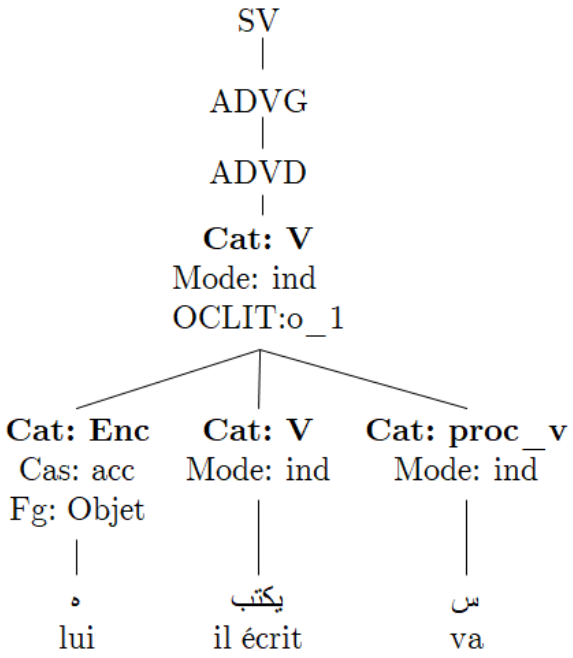


FIGURE 4.13: Arbre dérivé de la phrase (Il l'écrira)

Ainsi, pour la forme agglutinée "سيكتبه" (il l'écrira), illustrée par la figure 4.13, nous pouvons définir dans une même structure de traits que le proclitique "س" (sa) est une particule du futur (grâce au trait pos : proc_v). Ce genre de particules ne pourra s'attacher qu'au verbe au mode indicatif. Nous pouvons remarquer que le mode du verbe "كتب" (écrire) auquel cette particule s'est attachée est bien le mode indicatif ""يكتب (il écrit). Enfin, l'enclitique attachée à la fin du verbe "ه" (hou) représente l'objet du verbe (trait fg : objet1) dont le cas accusatif (trait cas : acc).

4.2.2.4 Omission du sujet

ArabTAG V2.0 couvre les phrases ayant un sujet elliptique et propose les modèles correspondants pour les représenter. La figure 4.14 montre deux phrases ayant la même signification : il dort.

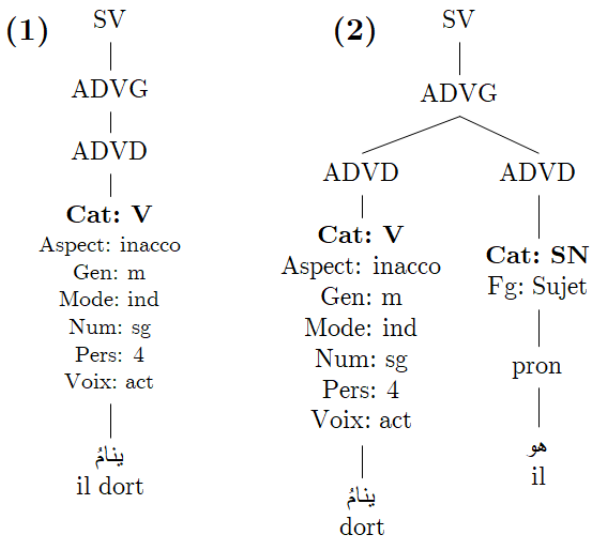


FIGURE 4.14: Les arbres dérivés de (Dort) et (Il dort)

Le modèle (1) représente l'arbre dérivé de la phrase "ينامُ" composé d'un verbe et d'un sujet elliptique. Tandis que le modèle (2) représente l'arbre syntaxique de la phrase (2) "هو ينامُ" composé d'un pronom et d'un verbe.

4.2.2.5 Structures enchâssées

La reconnaissance des structures enchâssées représente l'un des points forts de TAG. En effet, la représentation de ce phénomène est possible avec TAG grâce à l'opération d'adjonction. Cette opération permet d'insérer une structure complète dans une autre structure, rendant ainsi la représentation de l'enchâssement très naturelle. De plus, elle met en évidence la récursivité en permettant d'insérer plusieurs structures dans la même phrase. L'adjonction d'un arbre auxiliaire à lui-même ou à un autre arbre élémentaire met en valeur la récursivité de la langue naturelle.
Considérons la phrase suivante "التلميذ تسلّم الجائزة" (L'étudiant a reçu le prix) illustrée dans la figure 4.15. Nous pouvons insérer un syntagme subordonnée "الذي حقق النجاح" (qui a atteint le succès) entre le sujet "التلميذ" (l'étudiant) et son verbe "تسلّم" (a reçu). La phrase résultante est la suivante : "التلميذ الذي حقق النجاح تسلّم الجائزة" (L'étudiant qui a atteint le succès a reçu le prix).

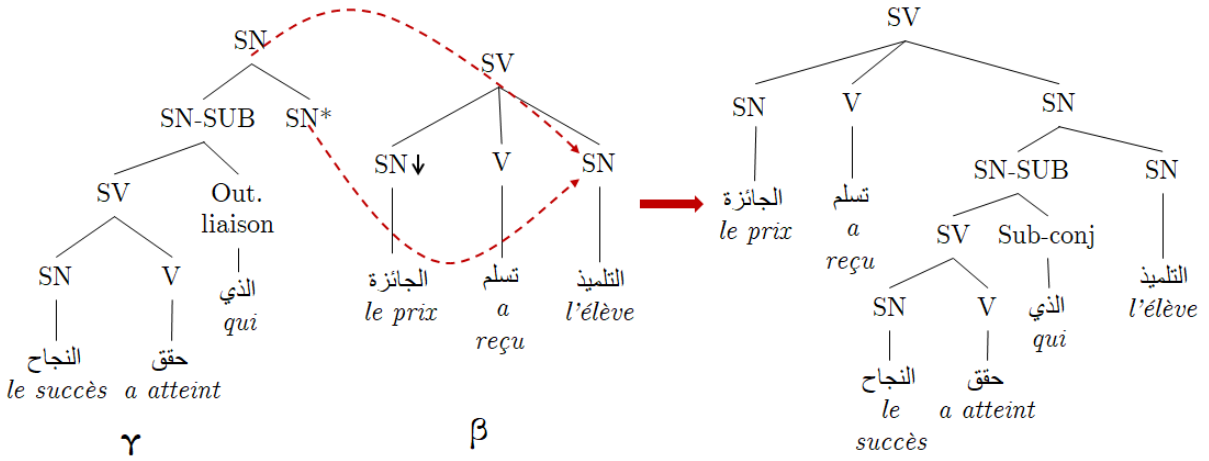


FIGURE 4.15: Exemple d'enchâssement dans TAG

4.2.2.6 Dépendances croisées

Le formalisme TAG permet de représenter des règles de grammaire (par des arbres élémentaires) de profondeur quelconque, supérieure ou égale à 1. Ainsi il est possible de définir un domaine de localité étendu. Ce domaine de localité offre la possibilité de décrire une représentation locale des dépendances syntaxiques. En effet, grâce à l'opération d'adjonction, il est possible de représenter des structures de phrases complexes telles que des phrases contenant des dépendances croisées. Ce phénomène se produit lorsque des relations de dépendance entre deux séries de mots se croisent.

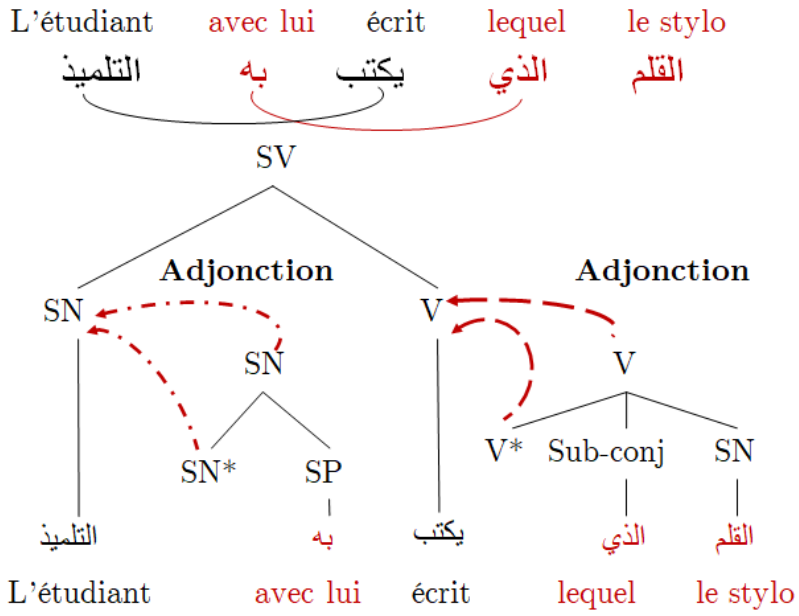


FIGURE 4.16: Exemple de gestion des dépendances croisées dans TAG

La figure 4.16 montre l'exemple de la phrase "القلم الذي يكتب به الطالب" (le stylo avec

lequel l'étudiant écrit). La gestion des dépendances croisées est réalisée en utilisant deux opérations d'adjonction dans la structure arborescente (SV).

4.2.2.7 Règles d'accord

Nous rappelons les nombreuses règles d'accord que nous avons présentées dans le chapitre 2 : l'accord entre l'adjectif et le nom dans un syntagme (définition, genre, nombre et cas), entre le sujet et le verbe d'une phrase verbale et enfin entre le thème et le propos dans une phrase nominale.
Ces règles sont gérées dans la grammaire ArabTAG V2.0 au moyen des traits morphosyntaxiques spécifiques aux nœuds impliqués dans l'accord. En utilisant ces traits, il nous a été possible de définir les contraintes appropriées pour assurer tous ces accords dans notre description méta-grammaticale.
Les traits morphosyntaxiques intervenant dans les règles d'accord sont décrits dans le tableau suivant (voir annexe A pour l'intégralité des traits définis dans notre méta-grammaire) :

Trait	**Description**	**Valeurs possibles**
Cat	Indique la catégorie du nœud au sein de la structure élémentaire	v, sv, sn, sp, p, adj, advg, advd, adv, inter, pinter, pn, circ, subor, subor_p, pactif, ppassif, comp, qual, nverbal, proc_v, cc, v_e, v_c, enc, proc, proc_def, proc_c, mushtaq, exc, app
SubCat	Indique la sous-catégorie du nœud au sein de la structure élémentaire	dem, pron, sn_sub, sn_subp,sn_ann, nom_prop, sn_sem_pac, sn_sem_ppas, sn_sem_comp, sn_sem_qual, sn_sem_nv, sn_adj, sn_def, sn_com, sn_dem
Fg	Indique la fonction du nœud au sein de la structure élémentaire	sujet, objet1, objet2, sujet_adj
FgType	Spécifie le type de la fonction du nœud au sein de la structure élémentaire	direct, indirect, pnv, non, interrogative, exclamative, appel
Gen	Spécifie le genre de la structure	M : le genre masculin, F : le genre féminin.
Nombr	Spécifie le nombre de la structure	sg : singulier, dl : dual, plr : pluriel.
Cas	Présente la voyelle de fin du mot	N : nominatif, A : accusatif, G : génétif.
Pers	Spécifie la personne	De 1 à 13

Def	Lorsque le nom est précédé par "ال"	+ : indique que le nom est défini (إسم معرّف) - : indique que le nom est indéfini (إسم غير معرّف)
Enc	Indique la présence ou non d'une enclitique liée au nœud en question	+ : présence d'une enclitique - : absence d'une enclitique
Humain	Indique si le nœud au sein de la structure élémentaire est humain ou non	+ : humain - : non humain
Anime	Indique si le nœud au sein de la structure élémentaire est animé ou non	+ : aimé - : non animé

TABLE 4.1: L'ensemble de traits morphosyntaxiques d'ArabTAG V2.0 intervenant dans les règles d'accord

- L'accord entre le verbe et le sujet :

Les arbres du sujet contiennent des équations d'accord qui lient les traits d'accord entre le nœud du verbe et son sujet. Lorsqu'un verbe précède le sujet, le verbe est au singulier mais s'accorde avec son sujet en genre. Cela se représente dans ArabTAG V2.0 de la manière suivante :

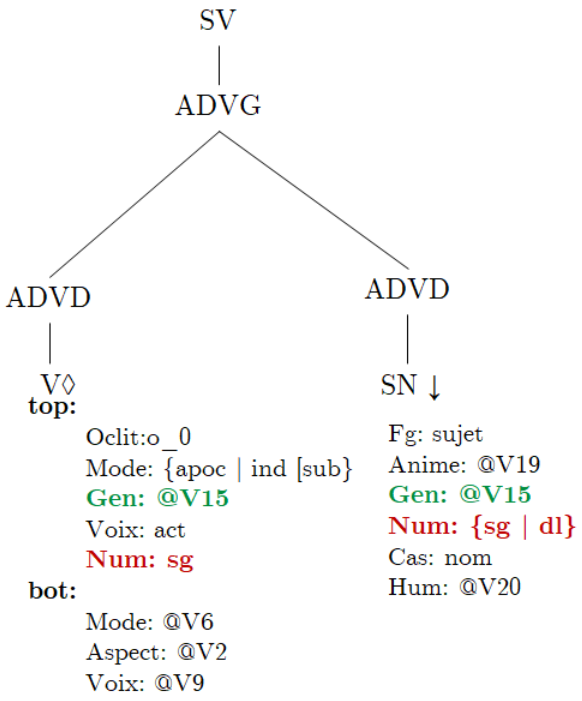


FIGURE 4.17: Accord entre le verbe et le sujet lorsque le verbe précède le sujet

L'accord en genre est réalisé avec l'équation d'égalité (dans la méta-grammaire) entre

les deux traits du genre pour le nœud du verbe et celui du sujet (?V_gen= ?SN_gen) et le verbe est toujours au singulier ?V_num=sg.
Ensuite, lorsque le sujet précède le verbe les deux cas distingués sont représentés de la manière suivante (figure 4.18) : (1) Si le sujet est non-humain (?Hum=h_0) pluriel(?SN_num=plr), le verbe s'accorde au féminin singulier avec ces équations ?V_num=sg et ?V_gen=f. Sinon (2) et (3), le verbe s'accorde en genre et en nombre avec ces équations d'égalité entre les traits du genre du nœud du verbe et du sujet ?V_gen= ?SN_gen et les traits du nombre ?V_num= ?SN_num.

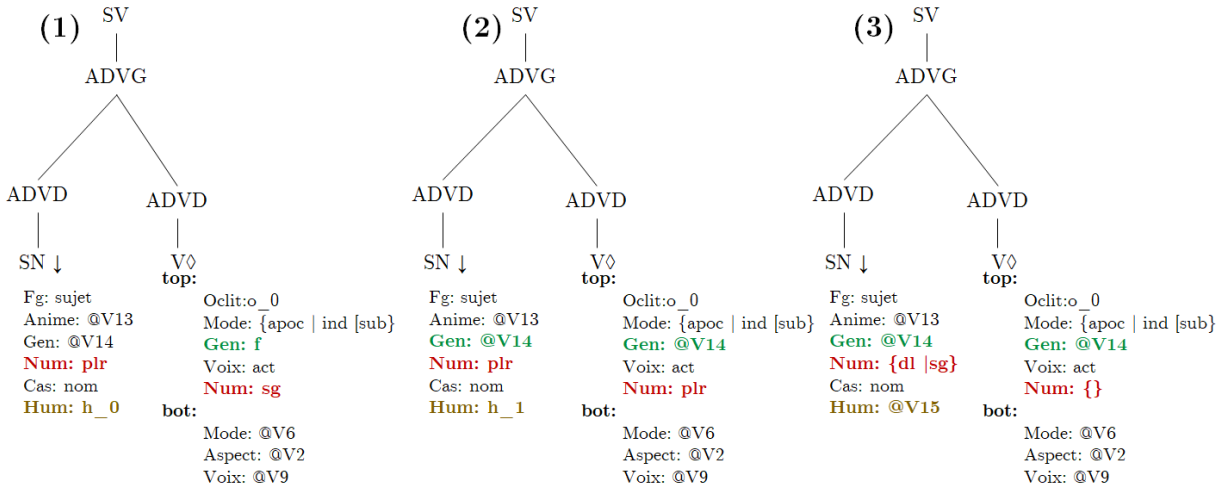


FIGURE 4.18: Accord entre le verbe et le sujet lorsque le sujet précède le verbe

- L'accord entre l'adjectif et le nom qu'il qualifie :
L'accord au sein d'un syntagme adjectival est assuré à l'aide des traits suivants : la définition (def), le genre (gen), le nombre (num) et le cas (cas). Ces contraintes sont définies dans les descriptions méta-grammaticales des structures arborescentes de ce syntagme, comme illustré par la figure 4.19.

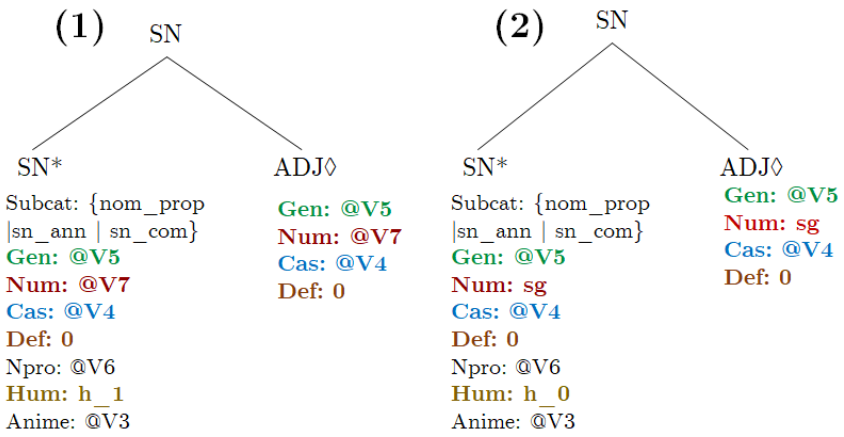


FIGURE 4.19: Accord entre le nom qualifié et son adjectif

Dans un syntagme adjectival, l'accord entre l'adjectif et le nom qu'il qualifie se fait

comme suit :
Pour obtenir un syntagme adjectival correct lorsque le nom qualifié est un être humain (1), non-humain singulier ou bien non-humain dual (2), l'adjectif (ADJ) doit s'accorder en genre, en nombre, en cas et en détermination avec le nom qu'il qualifie (SN).

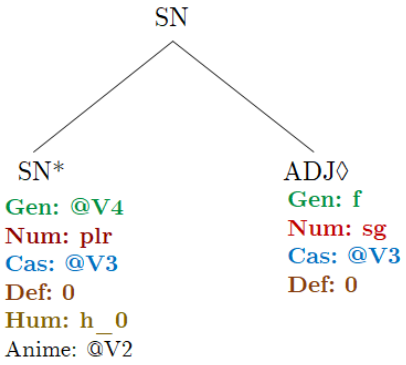


FIGURE 4.20: Accord entre le nom qualifié non humain pluriel et son adjectif

Cependant, si le nom qualifié est un non-humain pluriel (figure 4.20), l'adjectif est au féminin singulier et l'accord se fait uniquement en cas et en détermination.

4.2.3 Cycle de développement de la grammaire d'ArabTAG V2.0

Afin de vérifier la couverture grammaticale, nous avons mis en place un environnement de développement (script en langage Python) durant le processus de conception d'ArabTAG avec XMG. En plus de notre grammaire, nous avons défini, manuellement, des lexiques syntaxiques et morphologiques en suivant l'architecture en 3 couches du projet [XTAG, 2001].

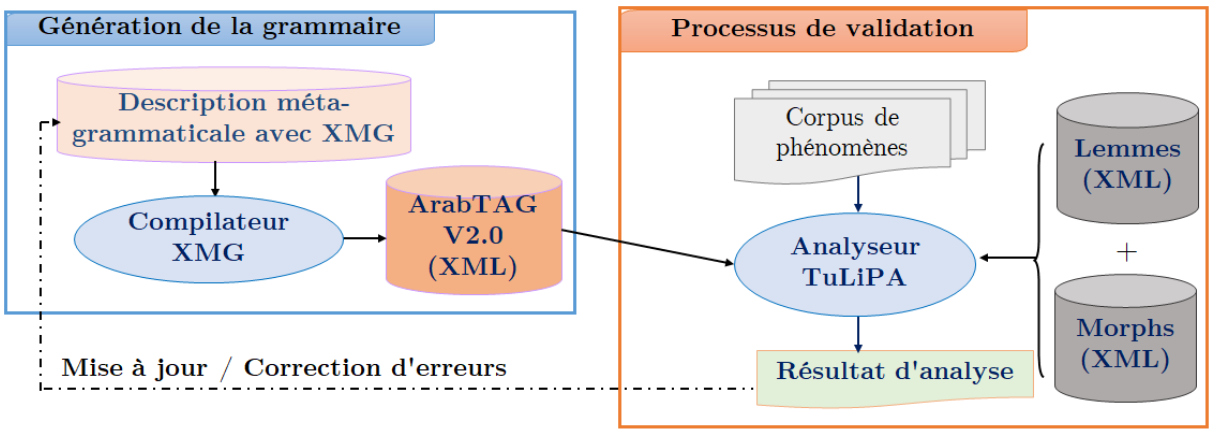


FIGURE 4.21: Architecture de validation d'ArabTAG V2.0

Le système XTAG se compose de trois sous-modules :

— Une base de schèmes classés en familles d'arbres élémentaires.

— Une base de lemmes où à chaque lemme est associé à une (ou plusieurs) famille(s) d'arbres.

— Une base morphologique dans laquelle chaque forme fléchie est associée à un lemme et à l'information morphosyntaxique appropriée.

Le but de ce test (figure 4.21) est d'évaluer à la fois la sous-génération et la surgénération des arbres élémentaires de notre grammaire. La grammaire générée doit être capable de reconnaître des phrases valables couvrant des phénomènes linguistiques en arabe (phrases décrites dans les manuels scolaires, les nouvelles en arabe, etc.) et de rejeter les phrases agrammaticales. Pour cette raison, nous avons construit un corpus de test dit : corpus de phénomènes.
A chaque nouveau phénomène syntaxique inclus dans ArabTAG V2.0, le corpus de phénomènes est enrichi manuellement avec des phrases grammaticales et agrammaticales. Chaque phrase de ce corpus, est associée au nombre d'analyses syntagmatiques attendues (0, 1 ou plus). A titre d'exemple, un extrait de ce corpus est illustré par la figure 4.22.

```
gloss    : aime ali fatima
text     : يحبُ عليٌ فاطمةُ
axiom    : sv
expected: 1
tags     :
  - transitif
  - sujet-nom-propre
  - objet-nom-propre
  - aime-ali-fatima
---
gloss    : aime ali
text     : يحبُ عليٌ
axiom    : sv
expected: 0
tags     :
  - transitif
  - sujet-nom-propre
  - objet-manquant
  - aime-ali
```

FIGURE 4.22: Extrait du corpus de phénomènes pour traiter des phrases avec un verbe transitif

Nous avons utilisé les deux phrases de cet extrait pour vérifier la couverture des phrases avec des verbes transitifs dans la grammaire générée. Ainsi à la fin de l'analyse syntaxique, nous nous attendons à obtenir un résultat pour la première phrase "يحبُ عليٌ فاطمةً" (aime Ali Fatima) (expected :1), qui est correcte, et aucun résultat pour la deuxième phrase "يحبُ عليٌ" (aime Ali) qui est incomplète (expected :0).
Le corpus de phénomènes compte 212 exemples de phrases (150 phrases grammaticales et 62 phrases non grammaticales). Il contient 134 phrases verbales, 45 phrases nominales, 32 syntagmes nominaux et un syntagme prépositionnel. Les exemples de phrases et syntagmes agrammaticaux ont été ajoutées pour vérifier si la grammaire peut renvoyer des arbres syntaxiques incorrectes.
Le tableau 4.2 résume les différents phénomènes couverts par notre grammaire :

Phénomènes	**Nombre de phrases/syntagmes**
Forme active	123
Adverbe	6
Forme agglutinée des mots	26
Règles d'accord	25
Complément circonstanciel	9
Verbe Ditransitif	67
Sujet elliptique	17
Structure enchâssée	11
Ordre semi-libre des mots	44
Phrase interrogative	10
Verbe intransitif	29
Forme passive	11
Verbe transitif	38

TABLE 4.2: Phénomènes couverts par le corpus

4.2.3.1 Répartition des modèles d'arbres générés dans ArabTAG V2.0

Jusqu'à présent, nous avons généré 1074 arbres à partir d'une description faite de 29 classes (soit 29 fragments d'arbres ou règles de combinaison). La distribution en familles de ces structures est illustrée par la figure 4.23.

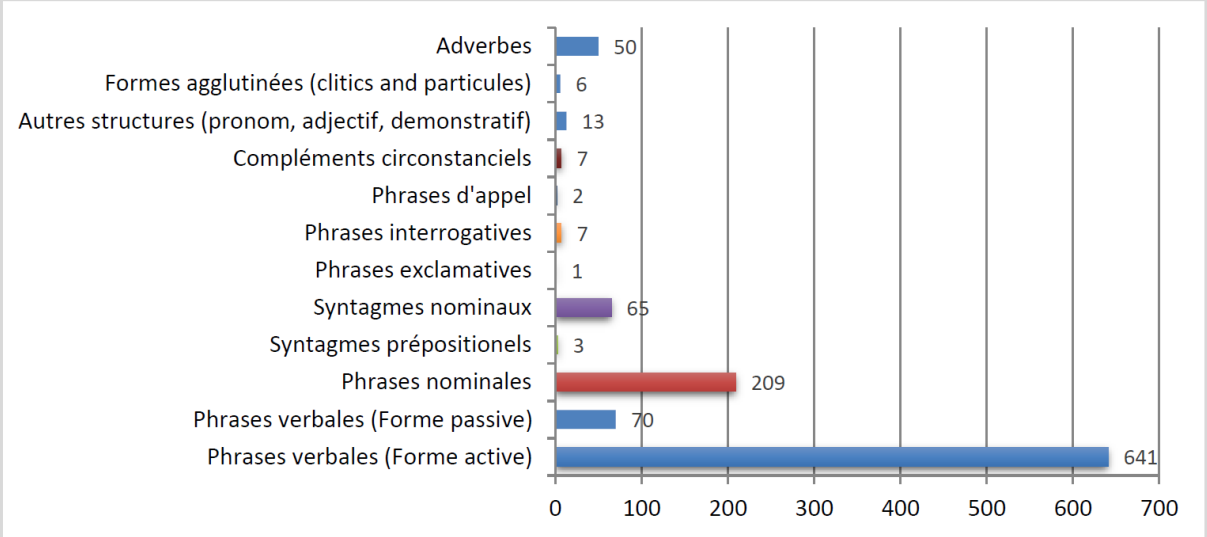


FIGURE 4.23: Distribution des modèles d'arbres dans ArabTAG V2.0

Grâce au corpus de phénomènes, nous avons pu vérifier que la grammaire générée couvre les structures déjà couvertes par la première version d'ArabTAG à savoir les phrases verbales (forme active et passive), phrases nominales, syntagmes nominaux, les syntagmes subordonnés et syntagmes prépositionnels. La couverture de la grammaire a été

également étendue, par rapport à la première version d'ArabTAG, en ajoutant des arbres élémentaires pour la représentation des phrases à sujet ellipse ainsi que des compléments supplémentaires tels que les compléments circonstanciels de temps, compléments circonstanciels de lieu et les adverbes. En plus ArabTAG V2.0 traitent des formes plus complexes des phrases ainsi que différents phénomènes de la langue arabe, tels que l'ordre semi-libre des mots, les formes agglutinées, les structures enchâssées, les dépendances croisées ainsi que les règles d'accord de l'arabe.
Néanmoins, les résultats mesurés durant ce processus de vérification ne nous permettent pas de conclure sur la qualité de la grammaire. Le corpus de phénomènes que nous avons défini manuellement est limité. Nous avons évalué notre grammaire en utilisant un corpus de plus grande taille sur des textes réels. Cette évaluation sera détaillée dans le dernier chapitre.

4.3 Intégration de la dimension sémantique dans la méta-grammaire

Afin d'étendre notre méta-grammaire et de produire une grammaire TAG à portée sémantique, nous avons pensé à associer aux familles des arbres décrites des cadres sémantiques. Nous nous sommes basés sur la théorie du "linking" [2] [Levin, 1993], [Kasper, 2008] selon laquelle le verbe permet d'exprimer dans la plupart des cas la sémantique d'un évènement ainsi que la relation entre ses participants. Par exemple, lorsque l'acteur du verbe est présent dans une phrase, il est souvent en position sujet (au cas nominatif). Ce genre de composant peut avoir le rôle d'"AGENT". Ainsi, la fonction grammaticale permet d'indiquer le rôle à attribuer. En d'autres termes, nous pouvons considérer que le prédicat verbal permet de constituer un cadre sémantique en sélectionnant l'ensemble des rôles sémantiques de ses participants. Certains de ces rôles sont obligatoires et déterminent la présence ou non de certaines fonctions grammaticales.
L'étiquetage de rôles sémantiques (Semantic Role Labeling : SRL) permet d'associer automatiquement les rôles sémantiques à chaque argument du prédicat d'une phrase. Cette tâche est utile pour diverses applications du domaine TALN tels que les systèmes de traduction automatiques [Liu and Gildea, 2010], les systèmes questions-réponses [Pizzato and Mollá, 2008], [Maqsud et al., 2014] ou encore les systèmes d'extraction de l'information [Christensen et al., 2011], [Fader et al., 2011]. Plusieurs de ces approches utilisent les ressources PropBank [Kingsbury and Palmer, 2003] et FrameNet [Baker et al., 1998] afin de définir le prédicat, les rôles utilisés lors de l'étiquetage ainsi que l'ensemble de test pour l'apprentissage automatique. En ce qui concerne l'arabe nous pouvons citer les travaux de [Diab et al., 2008] qui utilisent les machines vectorielles [Vapnik, 1998] et [Meguehout et al., 2017] qui se basent sur le raisonnement à partir de cas pour réaliser l'étiquetage sémantique.

Dans notre approche, nous ne nous sommes pas limités à l'étiquetage de rôles sémantiques. En effet, notre idée consiste à décrire, au moyen de XMG2, les différents modèles de cadres sémantiques (cadres pour les prédicats et cadres élémentaires) compatibles avec les familles d'arbres syntaxiques d'ArabTAG V2.0. L'unification de ces cadres est contrô-

2. C'est la mise en relation d'une structure en rôle sémantique avec une structure syntaxique.

lée par une hiérarchie de types et de contraintes que nous avons implémentée au sein de notre méta-grammaire. Quant aux rôles sémantiques, ils seront précisés au niveau du prédicat (qui est le verbe). Le cadre de la phrase est ensuite construit au fur et à mesure de l'analyse syntaxique en unifiant les cadres sémantiques élémentaires de ses composants syntaxiques par l'intermédiaire d'une interface syntaxe-sémantique.

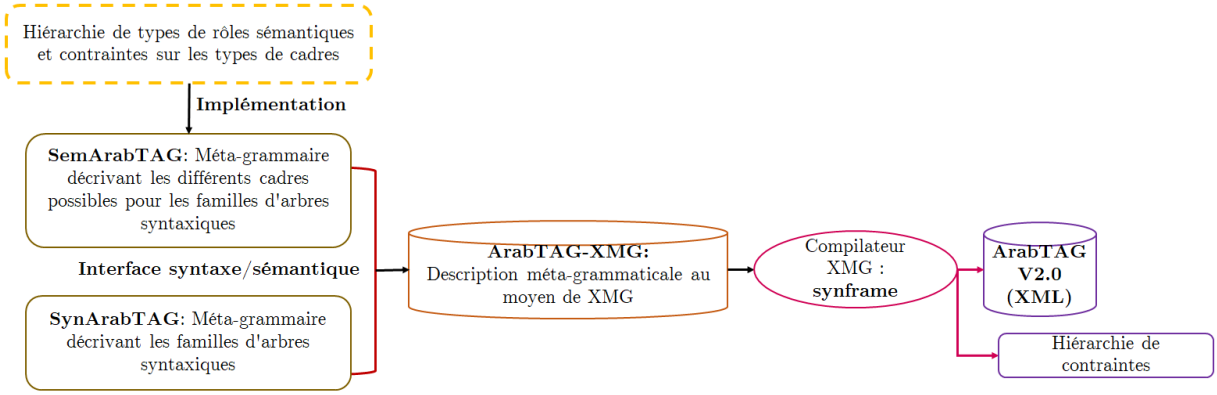


FIGURE 4.24: Processus de génération semi-automatique d'ArabTAG V2.0 avec une dimension sémantique

La figure 4.24, illustre les étapes que nous avons suivies pour réaliser notre approche :

1. Au niveau syntaxique des familles de classes décrites par la méta-grammaire (SynArabTAG), nous avons défini les arguments du prédicat (verbe). Rappelons que chacune de ces familles regroupent les arbres ancrés par un verbe et un nœud (de substitution) pour chaque argument du prédicat.
2. Au niveau sémantique (SemArabTAG), nous avons défini les modèles des cadres sémantiques correspondants à chaque famille d'arbres définies dans le niveau syntaxique. La dimension <frame> permet de décrire un cadre sémantique à l'aide de structures de traits typées.
3. Toujours au niveau de SemArabTAG, nous avons implémenté une hiérarchie des rôles sémantiques ainsi que les contraintes sur les types de cadres.
4. Le lien entre les éléments des cadres sémantiques et les constituants syntaxiques est établi à l'aide de l'interface syntaxe-sémantique en utilisant la dimension <iface>.
5. Le compilateur synframe de XMG2 génère la nouvelle version d'ArabTAG V2.0 (à partir d'ArabTAG-XMG qui réunit les deux descriptions méta-grammaticales) avec un fichier correspondant à la hiérarchie de contraintes implémenté précédemment.

Ce processus d'intégration de la dimension sémantique dans la méta-grammaire est détaillé dans la suite de ce chapitre. Mais, tout d'abord, nous commençons par la présentation de la hiérarchie de contraintes de types que nous avons implémentée dans SemArabTAG.
En effet, afin d'alimenter notre grammaire avec les rôles sémantiques, nous avons étudié les ressources lexicales disponibles permettant de réaliser cette tâche. Grace à cette étude, nous avons établi une hiérarchie de contraintes de types qui permettra de considérer les restrictions éventuelles lors de l'unification des cadres durant la phase de l'analyse sémantique.

4.3.1 Hiérarchie de contraintes de types implémentée dans SemArabTAG

La sémantique des cadres a fait l'objet de plusieurs applications et projets accessibles en ligne dont le projet de l'Université de Berkeley, FrameNet [Baker et al., 1998]. Ce dernier a été fondé depuis une dizaine d'année, dans le but de constituer une ressource linguistique basée sur les cadres sémantiques de Fillmore pour l'anglais.
La base de données construite dans le cadre de ce projet met en relation les cadres (frames) avec leurs Frame Elements (FE) et leurs unités lexicales (Lexical Units, LU). Les FE sont les rôles sémantiques associés de manière unique à chaque cadre. Nous distinguons deux types de FE : les core FE qui sont les rôles obligatoires et les non core FE qui sont les rôles généralement descriptifs et non spécifiques au cadre. Quant aux unités lexicales, elles désignent un mot ou groupe de mots qui évoquent le(s) cadre(s). Chaque unité lexicale est illustrée par des exemples de textes annotés manuellement.
Par exemple, pour un verbe "poursuivre", son cadre sémantique peut avoir plusieurs unités lexicales telles que : "chasser", "conduire", "courir", etc. Ses rôles sémantiques obligatoires sont (AGENT, THEME) et il peut avoir les rôles descriptifs suivant (LOCATION, MODE, DURATION, etc.).
La base de données lexicale de FrameNet (pour l'anglais) contient à ce jour 1224 cadres sémantiques et 13640 unités lexicales. Ce projet a servi de base à la construction de FrameNet pour d'autres langues notamment le danois, le Portugais, le japonais, le coréen, l'allemand et le suédois. Cependant et malgré quelques initiatives, tels que [Ghneim et al., 2009], une telle ressource pour l'arabe n'est pas encore établie. Nous avons donc entrepris d'exploiter une autre ressource qui propose un ensemble prédéfini de rôles et cadres sémantiques. Notre choix s'est porté sur la ressource lexicale ArabicVerbNet [Mousser, 2010].
Nous avons parcouru toutes les classes verbales d'ArabicVerbNet et nous avons regroupé les informations concernant les cadres décrits et les rôles de leurs participants. Grâce à ces informations nous avons établi deux hiérarchies : la hiérarchie des rôles sémantiques et la hiérarchie de types des cadres.

4.3.1.1 Hiérarchie des rôles sémantiques

La hiérarchie des rôles sémantiques que nous avons implémentée est illustrée par la figure 4.25. Au fait, nous nous sommes basés sur l'organisation des rôles proposée au sein de la ressource lexicale ArabicVerbNet qui elle-même s'inspire de VerbNet pour l'anglais [Kipper et al., 2008].
Grâce à cette hiérarchie, il devient plus simple d'attribuer les rôles généralisables pour les participants de différents types d'événements (tel qu'un acteur qui est l'instigateur de l'évènement), les spécifier (tel que Agent et cause) ou les distinguer. Cette distinction permet aussi de différencier les différentes classes verbales.

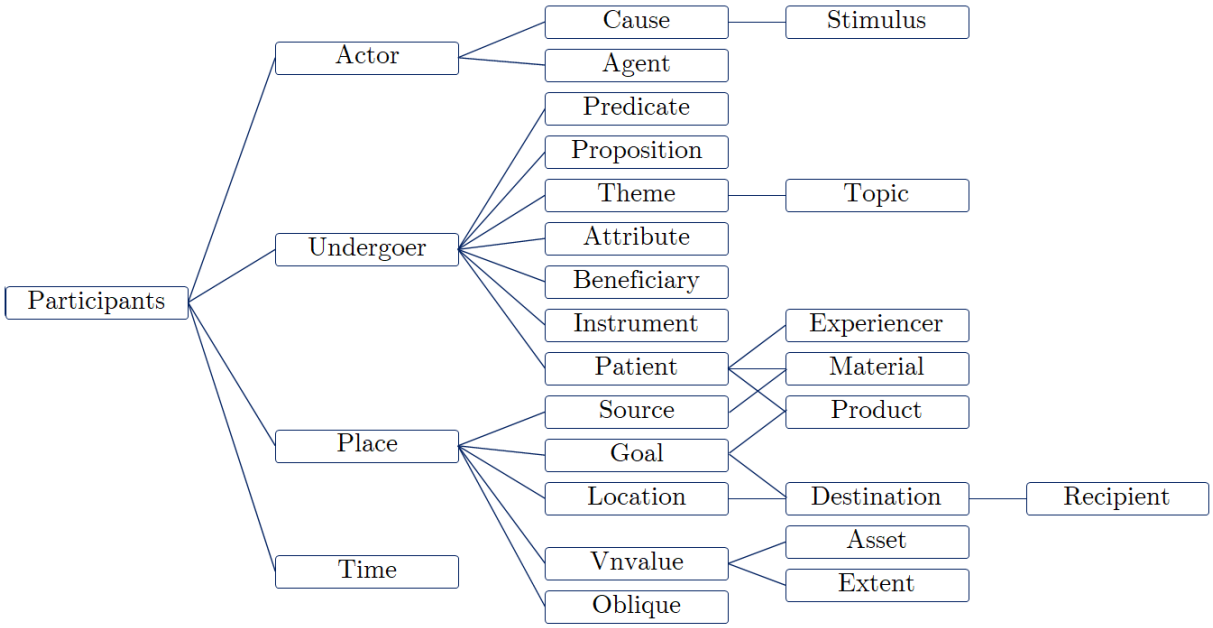


FIGURE 4.25: Hiérarchie des rôles sémantiques implémentée dans SemArabTAG

Le tableau 4.3 présente la description de quelques rôles de cette hiérarchie (une description de tous les rôles est fournie à l'Annexe B du manuscrit).

Rôle sémantique	**Description du rôle**
Actor	L'instigateur de l'évènement
Agent	C'est un acteur qui initie et réalise l'événement intentionnellement. Il existe indépendamment de l'événement.
Cause	C'est un acteur qui initie l'événement mais sans aucune intentionnalité ou conscience et qui existe indépendamment de l'événement.
Stimulus	Une cause dans un événement qui suscite une réponse émotionnelle ou psychologique (exemple : événements de perception)

TABLE 4.3: Description d'un extrait de rôles sémantiques

Comme l'indique le tableau ci-dessus, les rôles "CAUSE" et "AGENT" sont deux acteurs qui initient un évènement mais le premier le fait sans intention contrairement au second. Tandis qu'un "STIMULUS" c'est un type d'acteur non intentionnel spécifique qu'aux verbes de perception.
Ces rôles attribués aux participants de l'événement, représentent les attribus (frame Elements) qui seront affectés aux cadres des prédicats définies au sein de notre méta-grammaire au moment de l'analyse sémantique. Cependant, dans certain cas, un rôle doit obéir à une ou plusieurs contraintes de type.

4.3.1.2 Hiérarchie de types des cadres

Un verbe peut imposer un ensemble de restrictions à ses rôles d'argument. Par exemple, en exigeant qu'un rôle soit humain et /ou animé, etc.
Considérons les deux phrases suivantes avec leurs interprétations sémantiques :
(1) "يحبُ عليٌ فاطمةً" / Ali aime Fatima : EXPERIENCER عليٌ/ Ali +THEME فاطمةً/ Fatima.
(2) "يحبُ الكتابُ فاطمةً" / Le livre aime Fatima : EXPERIENCER الكتابُ/ le livre + THEME فاطمةً/ Fatima.

Le prédicat est le verbe "أحب" (aimer). Bien que les deux phrases soient syntaxiquement correctes la deuxième est sémantiquement incorrecte. Le sujet, qui est l'"EXPERIENCER" "الكتابُ" (livre), ne peut pas éprouver des sentiments envers un humain. Par conséquent, il est primordial de faire intervenir les contraintes spécifiées pour les rôles sémantiques afin de filtrer les phrases sémantiquement incorrectes. Après avoir examiné la classe verbale de "أحب" (aimer), nous avons remarqué que les restrictions imposées par cette classe sur l'"EXPERIENCER" c'est : animé et humain.
Nous avons exploité d'avantage les classes d'ArabicVerbNet et nous avons établi une deuxième hiérarchie (illustrée par la figure 4.26) qui concerne les types des cadres sémantiques élémentaires. Ces types peuvent aussi servir à imposer des restrictions sur les rôles sémantiques.

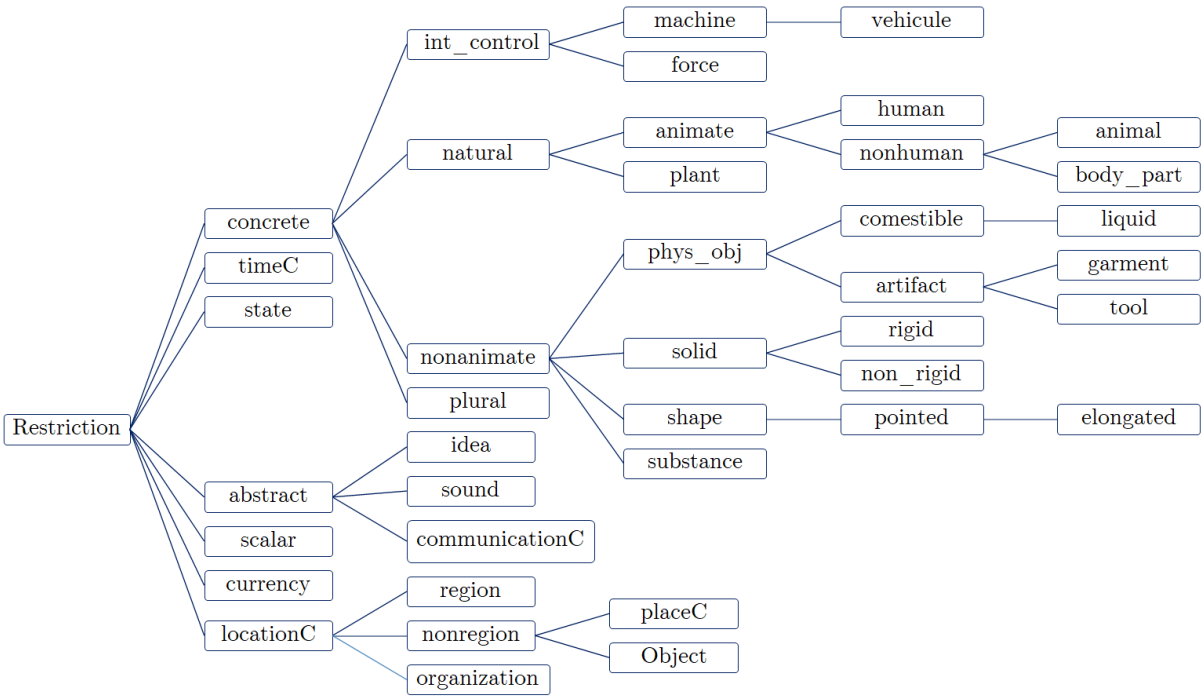


FIGURE 4.26: Hiérarchie de types des cadres sémantiques élémentaires implémentée dans SemArabTAG

Au moment de l'analyse sémantique, cette hiérarchie permettra d'optimiser la tâche

de l'étiquetage de rôles sémantiques et gérer les contraintes de type lors de l'unification des cadres.

4.3.2 Description méta-grammaticale des cadres sémantiques

Avec XMG2, nous avons décrit 27 cadres sémantiques sous format de structures de traits typées. Les modèles des cadres sémantiques décris dans notre méta-grammaire SemArabTAG sont divisés en deux catégories : les cadres sémantiques du prédicat et les cadres élémentaires.

4.3.2.1 Cadres sémantiques du prédicat

En respectant la modélisation hiérarchique des familles d'arbres des verbes (intransitif, transitif et ditransitif) nous avons adopté une organisation hiérarchique entre les descriptions des cadres sémantiques du prédicat (qui est le verbe). Cette organisation est schématisée par la figure 4.27.

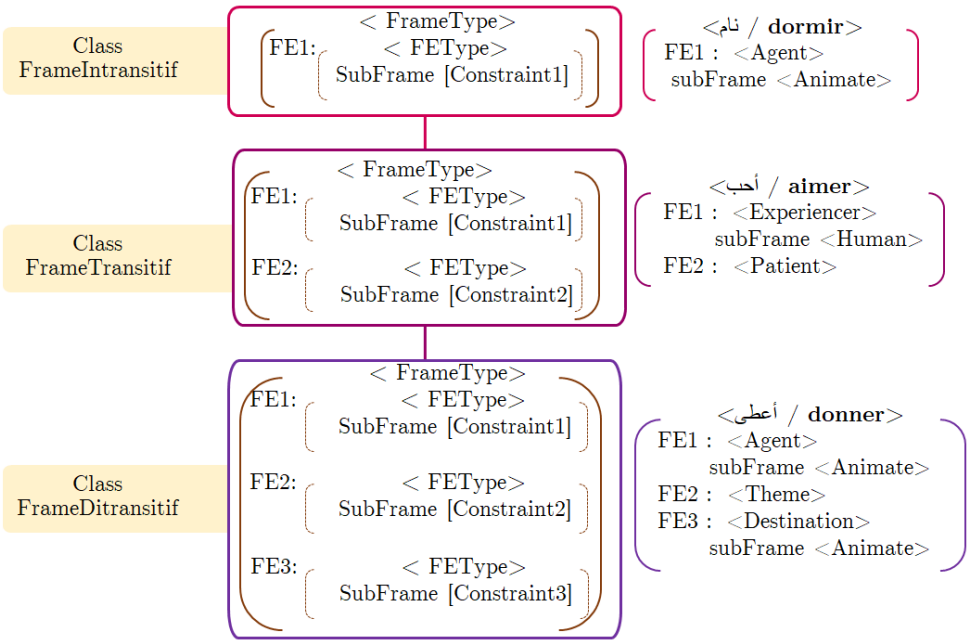


FIGURE 4.27: Hiérarchie des cadres sémantiques du prédicat dans SemArabTAG

Les cadres sémantiques décrits permettent de spécifier le type de l'évènement du prédicat avec la variable <FrameType>. Les rôles sémantiques correspondants aux participants de l'événement sont définis aux moyens des attributs FE. Ces derniers reçoivent comme valeur la variable <FEType> qui désigne le rôle sémantique et un cadre élémentaire <subFrame>. Ce cadre élémentaire devra respecter les restrictions éventuelles <constraint> sur le type du rôle sémantique.
Ainsi, un évènement exprimé par un prédicat verbal appartenant à une famille de verbe intransitif, par exemple "نام" (dormir), nécessite un seul participant désigné par l'attribut FE1. Ce dernier recevra la valeur du cadre représentant le rôle sémantique "AGENT" qui

devra avoir comme participant une entité animée.
De la même manière, un évènement exprimé par un prédicat verbal appartenant à une famille de verbe transitif (respectivement ditransitif), nécessite deux participants désignés par les attributs FE1 et FE2 (respectivement trois participants : FE1, FE2 et FE3). Par exemple, le cadre sémantique d'un verbe transitif "أحب" (aimer) est composé de deux participants. Le premier correspond au rôle sémantique "EXPERIENCER" (celui qui ressent l'évènement et qui doit être un humain) tandis que le second correspond au rôle "PATIENT" (celui qui subit l'évènement).
Enfin, le cadre d'un verbe ditransitif tel que "أعطى" (donner) est décrit pas trois participants : l' "AGENT" animé qui initie l'évènement, le "THEME" qui est au centre de l'évènement et enfin la "DESTINATION" qui désigne la location physique animée vers laquelle le "THEME" se dirige.
Afin de traiter les phrases à la forme passive, nous avons aussi décrit une hiérarchie des cadres sémantiques correspondants à leurs familles.

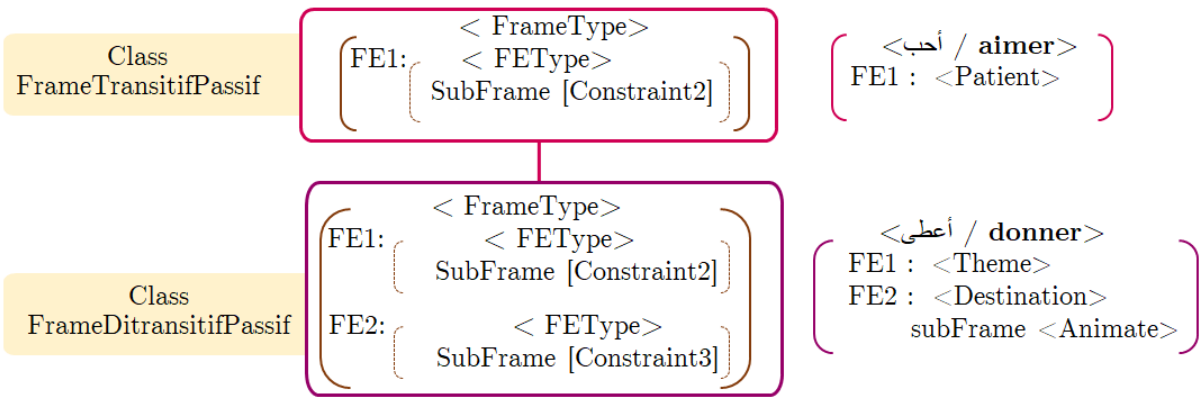


FIGURE 4.28: Hiérarchie des cadres sémantiques pour les familles passives dans SemArabTAG

Comme l'illustre la figure 4.28, le cadre sémantique du verbe "أحب" (aimer) sous sa forme passive est composé désormais d'un seul participant. Ce dernier correspond aux second rôle sous sa forme active "PATIENT". De la même manière, "أعطى" (donner) sous sa forme passive est décrit pas deux participants : le "THEME" et la "DESTINATION".
Les rôles sémantiques sont implémentés en tant que types <FEType> et non pas en tant qu'attributs. Nous avons adopté ce choix à la suite des limites relevées en définissant les rôles en tant qu'attributs. Ceci est illustré par la figure 4.29.
Dans le but de générer les cadres sémantiques possibles de "FrameTransitif" nous avons combiné les deux classes "FrameIntransitif" et "FrameTransitif_ ". Ces deux dernières définissent chacune l'ensemble des cadres possibles contenant un rôle sémantique. [3] Parmi les six cadres résultants de la conjonction, nous avons relevé la génération d'un cadre contenant un seul attribut "Patient". Le résultat est faux puisque ce cadre doit avoir deux participants dont le rôle est "Patient". En effet, lors de la conjonction, l'attribut

3. La liste des rôles possibles a été extraite de ArabicVerbNet

"Patient" de "FrameIntransitif" et l'attribut "Patient" de "FrameTransitif_" sont unifiés. En d'autres termes, il est impossible de générer un cadre contenant deux attributs avec la même valeur. La première alternative pour résoudre ce problème est de distinguer les attributs en les numérotant. Cependant la conséquence directe de cette distinction est l'augmentation du nombre possible des rôles sémantiques attribués et ainsi l'explosion des nombres des cadres générés. Dans l'exemple de la figure 4.29, nous avons délibérément limité le nombre des rôles possibles à trois dans "FrameIntransitif" et deux "FrameTransitif_ " mais lors de l'implémentation il faudra compter au moins une dizaine de rôles pour chaque classe.

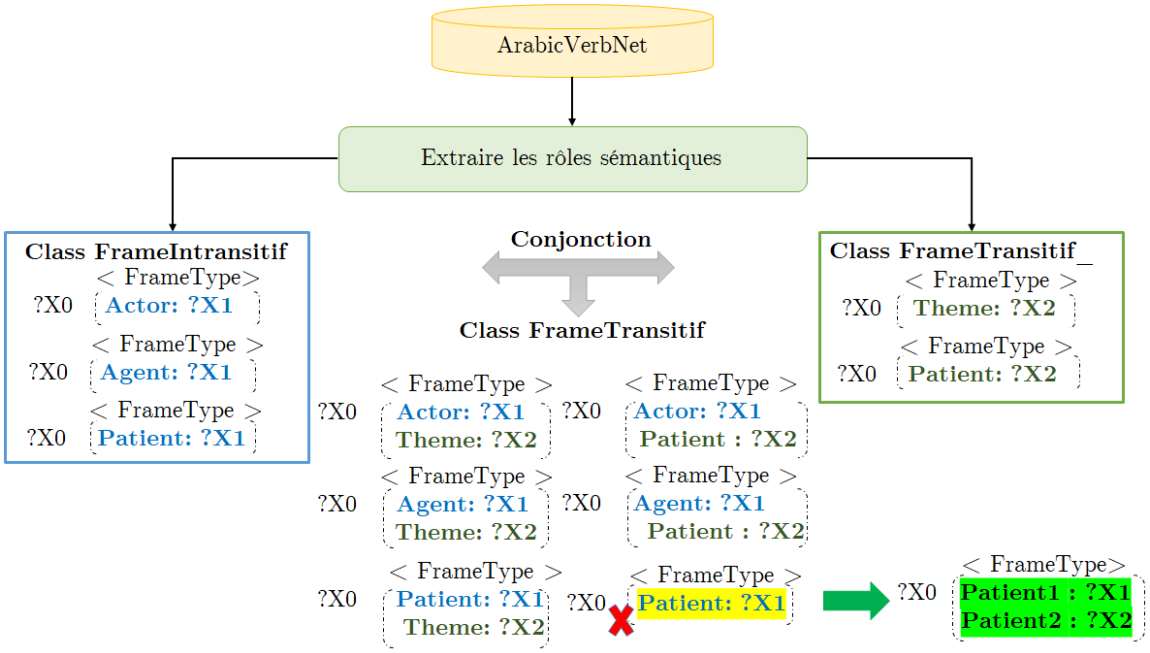


FIGURE 4.29: Problème détecté lors de définition des rôles sémantique en tant qu'attributs

Nous avons donc choisi une autre alternative en implémentant les rôles sémantiques en tant que types. Cette démarche nous a permis de définir les cadres du prédicat indépendamment de ses rôles sémantiques. Ainsi, nous avons uniquement un cadre sémantique pour chaque famille. De plus, nous pourrons obtenir des cadres avec des rôles de la même valeur. [4]

4.3.2.2 Cadres élémentaires

Nous avons décrit, au sein de notre méta-grammaire, les cadres élémentaires correspondants aux cadres des syntagmes nominaux. Lors de l'analyse sémantique, ces cadres vont s'unifier avec le cadre du prédicat (éventuellement avec d'autres cadres élémentaires) afin de construire le sens de la phrase.
Par exemple, un cadre élémentaire d'un syntagme nominal simple est décrit au sein de la classe FrameNomCommun. Le cadre élémentaire décrit permet de spécifier le type du syntagme ainsi que la valeur de son lemme.

4. La valeur du rôle est spécifiée au niveau du lexique

La figure 4.30 illustre l'arbre syntaxique du syntagme nominal "الشرطيُ" (le policier) ainsi que son cadre élémentaire correspondant.

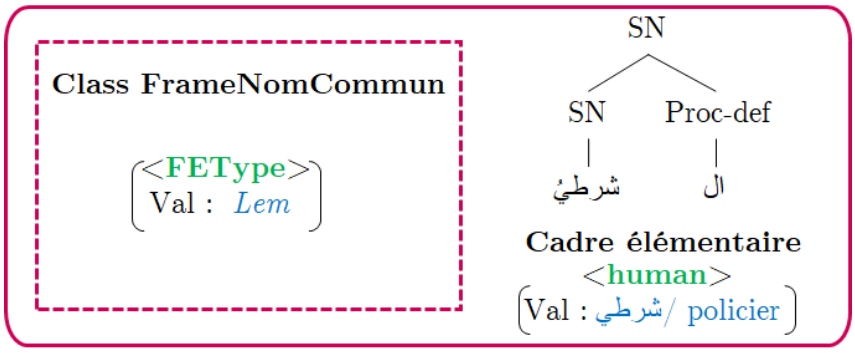


FIGURE 4.30: Exemple du cadre élémentaire pour un syntagme nominal simple défini dans SemArabTAG

Quant aux noms propres, leur cadre élémentaire est décrit au sein de la classe FrameNomPropre. Il décrit le type du nom propre ainsi que le lemme du nom.
La figure 4.31 illustre un exemple d'un cadre élémentaire du nom propre "علي" (Ali).

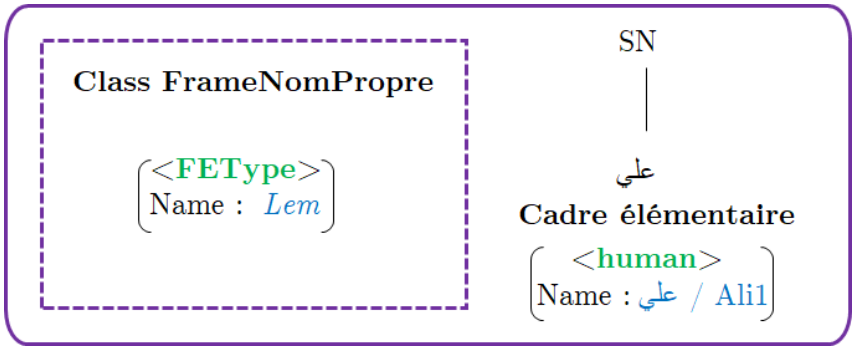


FIGURE 4.31: Exemple d'un cadre élémentaire pour un nom propre défini dans SemArabTAG

4.3.3 Construction de l'interface syntaxe-sémantique

L'interface syntaxe-sémantique que nous avons établi [Ben Khelil et al., 2018a], correspond à la définition d'une matrice de traits au moyen de la dimension <iface> pour chaque classe. Cette matrice permet d'associer un nom global (le trait) à une variable (la valeur du trait) ce qui permettra d'unifier les variables (à la suite d'une opération de substitution ou d'adjonction) du même nom global et faire la correspondance entre les arguments du prédicat et leurs rôles correspondants.
La figure 4.32 présente un exemple de l'interface syntaxe-sémantique entre une structure syntaxique d'une phrase verbale et son cadre sémantique correspondant. Le lien entre ces deux dimensions est assuré par les traits E (pour l'événement du prédicat) arg0 (pour

le premier argument) et arg1 (pour le deuxième argument). Les deux derniers traits reçoivent, dans la dimension sémantique, les valeurs des rôles sémantiques correspondants aux arguments du prédicat.

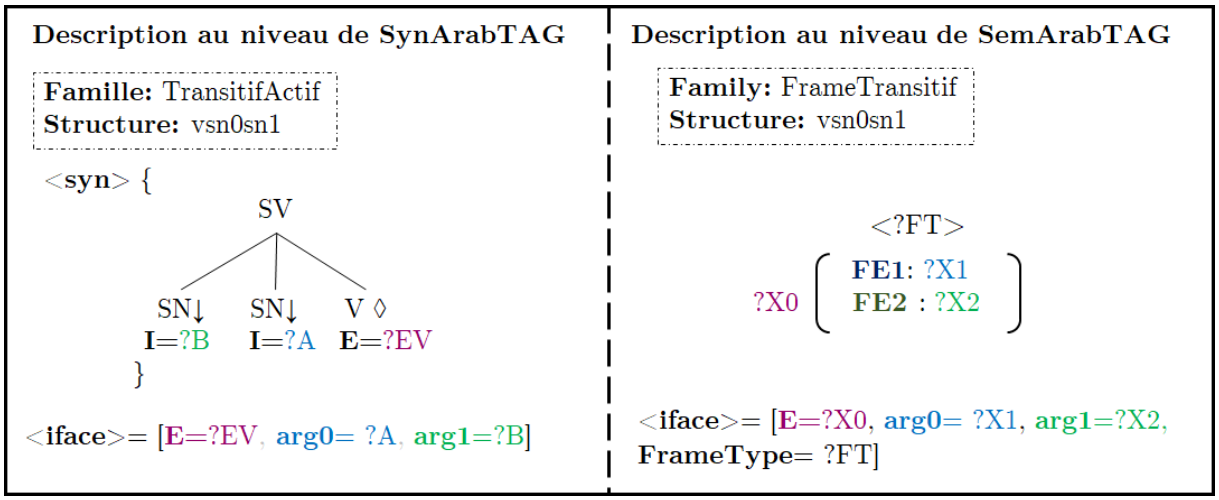


FIGURE 4.32: Description de l'interface syntaxe-sémantique au niveau de la méta-grammaire

Considérons l'exemple de la phrase suivante : "طاردَ الشرطيُ اللصَ" (le policier poursuit le voleur). En appliquant notre approche, le processus de construction du sens de cette phrase est réalisé comme suit :

1. Au niveau du lexique, à chaque élément de la phrase est associé son cadre sémantique et sa famille d'arbre syntaxique, de la manière suivante : Les cadres élémentaires (figure 4.33) sont attribués aux syntagmes : "الشرطيُ" (le policier) et "اللصَ" (le voleur)

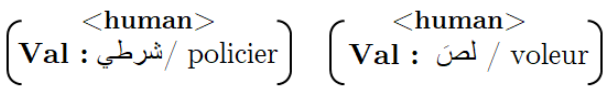


FIGURE 4.33: Cadres élémentaires attribués au (le policier) et au (le voleur)

Au niveau du cadre du prédicat les rôles sémantiques et leurs contraintes sont définis en faisant une correspondance entre le verbe "طاردَ" (poursuivre) et les classes d'ArabicVerbNet.

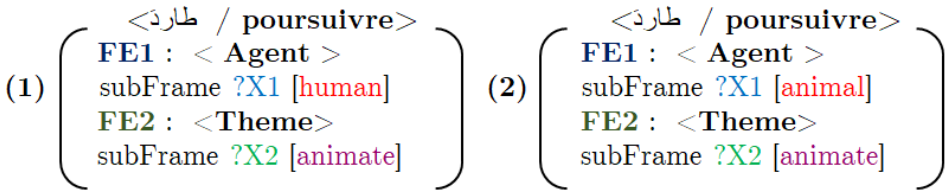


FIGURE 4.34: Les cadres sémantiques attribués au verbe (poursuivre)

A l'issue de cette correspondance les deux rôles attribués au prédicat sont : AGENT et THEME. Chacun de ces rôles admet ses contraintes de type. L'"AGENT" peut être "human" ou "animal" et le "THEME" est un "animate".
Ainsi le verbe "طاردَ" (poursuivre) se voit attribuer deux combinaisons de cadres sémantiques possibles illustrées par la figure 4.34.

2. L'élément I de la figure 4.35 représente l'interface syntaxe-sémantique. Cet élément permet le partage des variables de traits des nœuds avec les variables issues des cadres sémantiques.

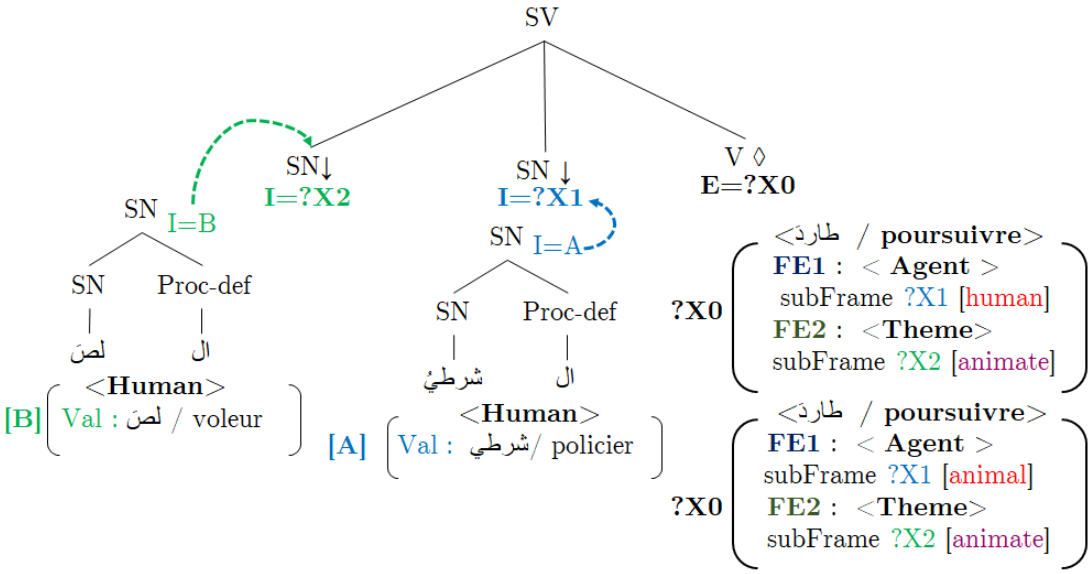


FIGURE 4.35: Processus de l'analyse syntaxico-sémantique de la phrase

3. Les opérations de substitution déclenchent les équations d'unification entre ces variables : [X1 = A] et [X2 = B].

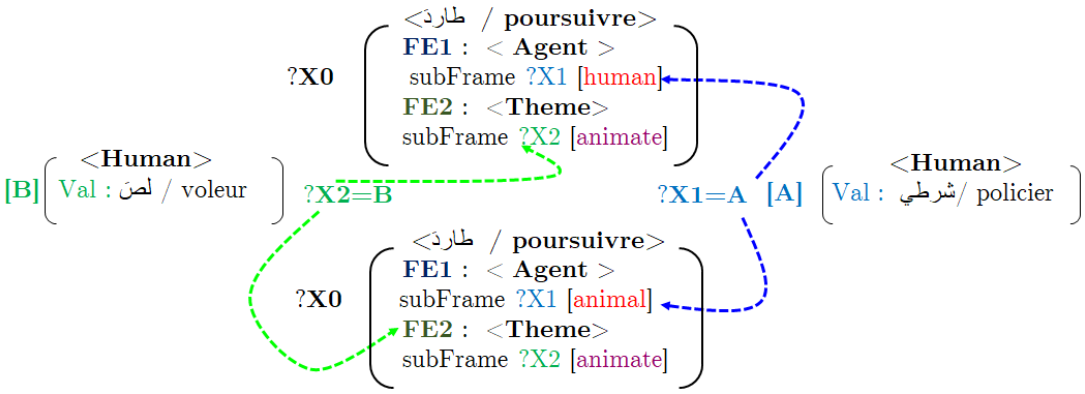


FIGURE 4.36: Déclanchement du processus d'unification des cadres sémantiques

4. Durant le processus d'unification, l'analyseur syntaxico-sémantique va devoir aussi traiter l'ambiguïté au niveau des cadres sémantiques du prédicat. Grace à la hiérarchie des contraintes sur le type, l'analyseur vérifie si les contraintes imposées par les

rôles "AGENT" (?X1) et "THEME" (?X2) sont unifiables avec le type "human" des deux cadres élémentaires [A] et [B].

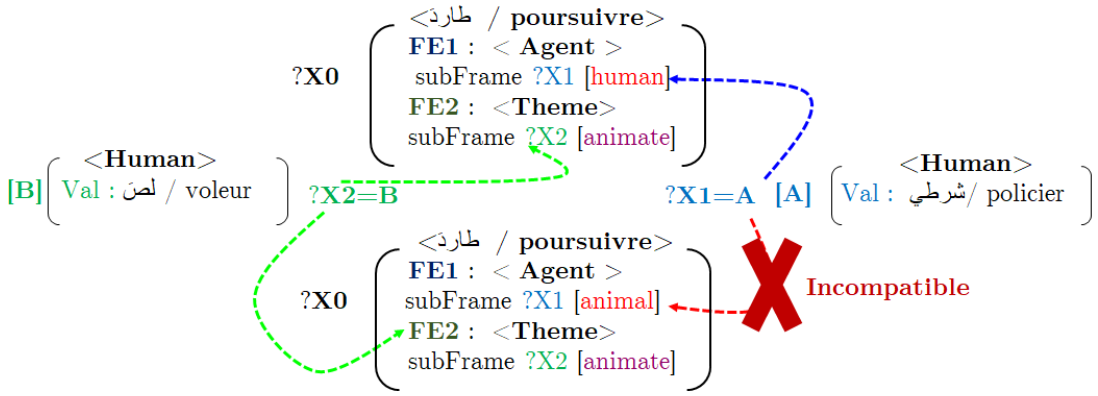


FIGURE 4.37: Processus d'unification des cadres sémantiques

La hiérarchie implémentée distingue le type "human" du type "animal" qui sont tous les deux des sous types de "animate". En d'autres termes, le type "humain" du cadre "اللصّ" (le voleur) peut s'unifier avec le type "animate" du rôle "THEME" des deux cadres sémantiques du verbe "طاردَ" (poursuivre). En revanche, le type du cadre "الشرطيُ" (le policier) est unifiable avec un seul cadre du verbe "طاردَ" (poursuivre). En effet, l'unification entre "human" et "animal" échoue, entrainant ainsi la résolution de l'ambiguïté gardant uniquement le premier cadre sémantique du prédicat.

5. À la fin de l'unification, les cadres élémentaires de "الشرطيُ" (le policier) et "اللصّ" (le voleur) sont insérés dans le cadre sémantique prédicat du verbe "طاردَ" (poursuivre) formant ainsi le cadre final représentant le sens de la phrase. Le résultat de l'analyse syntaxico-sémantique, à savoir l'arbre syntaxique et son cadre sémantique correspondant, est illustré par la figure 4.38.

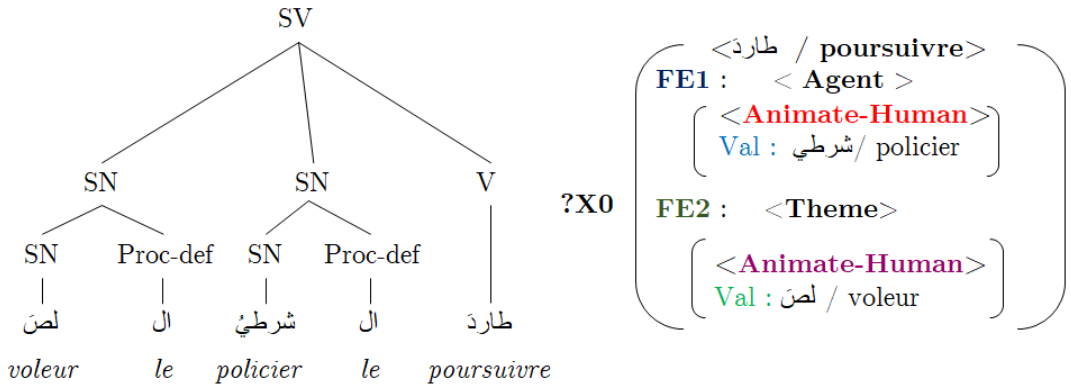


FIGURE 4.38: Résultat de l'analyse syntaxico-sémantique

4.4 Conclusion

Nous avons présenté dans ce chapitre notre approche qui propose un nouveau processus de génération semi-automatique d'une TAG baptisée ArabTAG V2.0, pour représenter la syntaxe et la sémantique de l'arabe moderne standard. Cette grammaire [5] a été réécrite en utilisant le langage de description méta grammatical XMG. Grâce à ce formalisme extensible, nous avons mis en place une représentation compacte de l'information grammaticale de l'arabe et ensuite nous avons établi l'ensemble des règles pour combiner ces fragments élémentaires d'information.
Cette grammaire décrit les structures principales des phrases de l'arabe : les phrases verbales (forme active et passive), les phrases nominales, les différents types des syntagmes nominaux et les syntagmes prépositionnels. De plus, elle traite aussi les différents phénomènes linguistiques tels que la variation des positions des éléments au sein des composants syntaxiques, les compléments supplémentaires, les règles d'accord et les formes agglutinées. En outre, ArabTAG V2.0 utilise les traits d'unification. Ces traits présentent des informations morphologiques, syntaxiques et syntaxico-sémantiques supplémentaires qui sont associées à un mot ou à un syntagme. En effet, nous avons étendu notre définition méta-grammaticale en intégrant l'information sémantique. Notre idée consiste à associer aux familles d'arbres élémentaires de la grammaire, une sémantique à base de cadres et d'inclure les rôles sémantiques et leurs contraintes à partir de la ressource ArabicVerbNet. Ceci a permis d'établir une correspondance entre arguments sémantiques et arguments syntaxiques par l'intermédiaire d'une interface syntaxe-sémantique permettant aux cadres sémantiques élémentaires de s'unifier lors de la composition syntaxique.
A présent, il nous faudra évaluer la qualité de couverture de notre grammaire générée. Pour effectuer cette évaluation, nous devons disposer d'un corpus de test plus riche et d'un ensemble d'outils d'analyse. Ce processus d'évaluation ainsi que les résultats sont présentés dans le chapitre suivant.

5. Accessible au lien suivant : https ://github.com/Cherifabk

Troisième partie

Évaluation

Chapitre 5

Évaluation d'ArabTAG V2.0

5.1 Introduction

L'étape d'évaluation est primordiale et doit prendre en compte un certain nombre de critères. Il est indispensable de s'interroger non seulement sur les ressources nécessaires à une telle évaluation mais aussi sur la procédure à adopter et les critères d'évaluation à utiliser. Durant la phase de validation d'ArabTAG V2.0, présentée dans le chapitre précédent, nous avons défini le lexique et le corpus de test manuellement. Ce dernier est constitué de 120 phrases dont 32 agrammaticales. Cependant, une telle évaluation utilisant un corpus limité ne nous permet pas de conclure sur la qualité de la grammaire. Nous avons donc évalué notre grammaire en utilisant un corpus arboré de grande taille.
Nous consacrons ce dernier chapitre à l'évaluation de notre approche. La première section souligne les objectifs que nous nous sommes fixés de cette évaluation. La section 2 est divisée en deux grandes parties. La première est consacrée à l'évaluation syntaxique de la couverture de la grammaire ArabTAG V2.0. Elle est subdivisée en sous sections pour la présentation du protocole de l'évaluation : corpus utilisé, le lexique construit, processus de l'analyse syntaxique et la présentation des résultats obtenus. Quant à la deuxième partie, elle est dédiée à l'évaluation sémantique. Au même titre que l'évaluation syntaxique, cette partie présente le corpus de test utilisé, le processus de test, la discussion des résultats obtenus ainsi que le rôle de l'analyse sémantique dans la désambiguïsation des représentations syntaxiques.

5.2 Objectifs de l'évaluation

Avant d'entamer la phase d'évaluation, nous devons faire le point sur ce que notre grammaire doit satisfaire. Notre travail de thèse a pour objectif final de mettre en place une grammaire pour décrire la syntaxe et la sémantique de l'arabe moderne standard. Celle-ci doit donc pouvoir :

— Distinguer les phrases valides grammaticalement telles qu'elles sont décrites dans l'arabe standard moderne(livres scolaires, romans arabes, etc.) et qui couvrent les phénomènes syntaxiques importants de l'arabe,

— Reconnaître les phrases agrammaticales,

— Fournir une description syntaxico-sémantique pour les phrases valides.

Les métriques classiques utilisées pour l'évaluation de l'analyse syntaxique et sémantique sont la précision et le rappel. La précision, permet de mesurer le bruit. [1] Elle représente le nombre de phrases grammaticales bien analysées par rapport au nombre total de phrases analysées. Quant au rappel, il est défini par le nombre de phrases grammaticales bien analysées par rapport au nombre total de phrases grammaticales présentes dans le corpus de test. Le rappel permet aussi de mesurer le silence. [2] Ces deux métriques (précision et rappel) sont combinées dans une mesure nommée F1-mesure (ou encore le f1-score). La quantification de ces critères se fait sur la base d'un corpus annoté manuellement, typiquement un corpus arboré.

5.3 Protocole d'évaluation et résultats

La difficulté majeure que nous avons rencontrée dans cette étape concerne les ressources nécessaires à l'évaluation. En effet, nous avons constaté que la disponibilité de ces ressources numériques est contrastée selon les langues. Nous avons exploré au préalable (voir chapitre 2) l'ensemble des trois corpus arborés pour l'arabe standard à savoir : Penn Arabic Tree Bank (PATB), Prague Arabic Dependency Treebank (PADT) et Columbia Arabic Treebank (CATiB).
Ces corpus n'adoptent pas le même format d'annotation. PATB repose sur les représentations à base des structures de constituants puisque les données sont étiquetées à trois niveaux : morphologique, syntaxique et grammatical. Tandis que PADT et CATiB suivent, plutôt, un format à base des structures de dépendances. Le format de ces dépendances est simple, étiqueté par les relations fonctionnelles des différents composants syntaxiques. Les textes inclus dans ces trois corpus sont des textes journalistiques. Cependant, nous voulons exploiter des textes littéraires renfermant des structures plus riches et plus représentatives de l'arabe et de ses phénomènes syntaxiques. En plus, tous ces corpus ne sont pas des ressources libres et leurs coûts sont élevés.
Face à ces contraintes, nous avons décidé de construire notre propre corpus de test. De plus, nous avons mis en place un protocole d'évaluation qui a permis de réaliser, en premier lieu, l'évaluation syntaxique et ensuite d'effectuer l'évaluation sémantique via une analyse syntaxico-sémantique d'un ensemble de phrases tests.

5.3.1 Évaluation de l'analyse syntaxique

Nous commençons dans cette section par la présentation des corpus de test utilisés pour l'évaluation syntaxique. Ensuite, nous détaillons chaque étape du processus de cette évaluation, avant de terminer par une discussion des résultats obtenus.

1. Bruit = 1- précision : mesuré lorsque des réponses non-pertinentes/non correctes appartiennent au résultat.

2. Silence = 1- rappel : lorsque les réponses pertinentes/ou correctes ne sont pas obtenues par le système alors qu'elles existent.

5.3.1.1 Corpus de test

La construction du corpus de test s'est faite manuellement par l'extraction de 1000 phrases, syntaxiquement correctes, du livre scolaire tunisien (niveau 8ième année). [3] Ce livre couvre un ensemble assez large des structures syntaxiques des phrases dans l'arabe tout en considérant les phénomènes suivants :

- Négation,
- Exclamation,
- Interrogation,
- Compléments optionnels,
- Structures verbales,
- Structures nominales,
- Forme active,
- Forme passive,
- Accord,
- Temps et aspect des verbes,
- Coordination,
- Enchâssement.

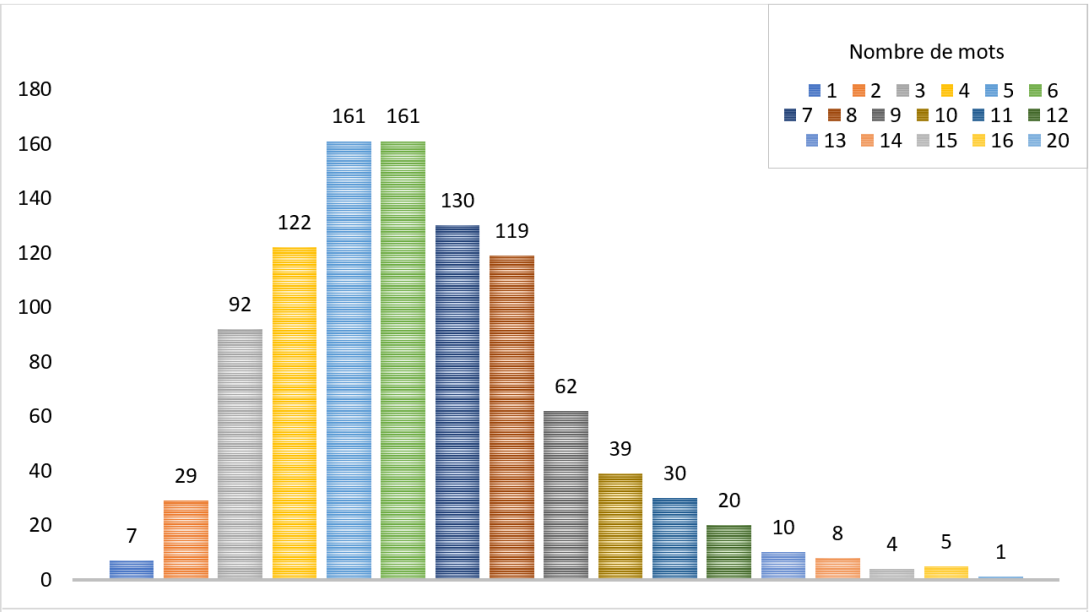


FIGURE 5.1: Répartition des phrases du corpus de test en fonction de leurs longueurs

Contrairement au corpus de phénomènes que nous avons utilisé lors du développement de ArabTAG V2.0 (voir section 1.3 du chapitre 4), ce nouveau corpus est plus riche, composé d'un ensemble de 650 phrases verbales et 350 phrases nominales extraites de textes réels couvrant ainsi plus de structures complexes et de phénomènes de l'arabe. Le nombre des

3. Equivalent à la 4ième année au collège en France.

mots de ces phrases varie entre 1 et 20 mots (voir figure 5.1).
Afin d'évaluer la capacité de notre grammaire à reconnaitre des phrases agrammaticales, nous avons transformé quelques phrases de notre corpus de test et nous avons créé 250 phrases grammaticalement incorrectes. Cet ensemble de phrases est constitué de 167 phrases verbales et 83 phrases nominales. Ces phrases couvrent les erreurs suivantes :

— L'accord incorrect entre le verbe et le sujet,
— L'accord incorrect entre le thème et le propos,
— L'accord incorrect entre le verbe d'existence ou de certitude et son thème ou propos,
— L'accord incomplète,
— Ordre incorrect au sein du syntagme,
— Ordre incorrecte des mots de la phrase.

5.3.1.2 Processus de l'évaluation syntaxique

Le processus d'analyse, présenté par la figure 5.2, a été réalisé à l'aide d'un outil que nous avons mise en place. Cet outil permet de :

1. Réaliser l'étiquetage morphosyntaxique de la phrase en entrée.
2. Générer les bases de lemmes et de morphes des mots étiquetés.
3. Exécuter l'analyseur syntaxique TuLiPA.
4. Stocker le résultat de l'analyse (arbre(s) résultat(s) étiqueté(s)) dans un nouveau fichier afin de construire un corpus arboré et étiqueté.

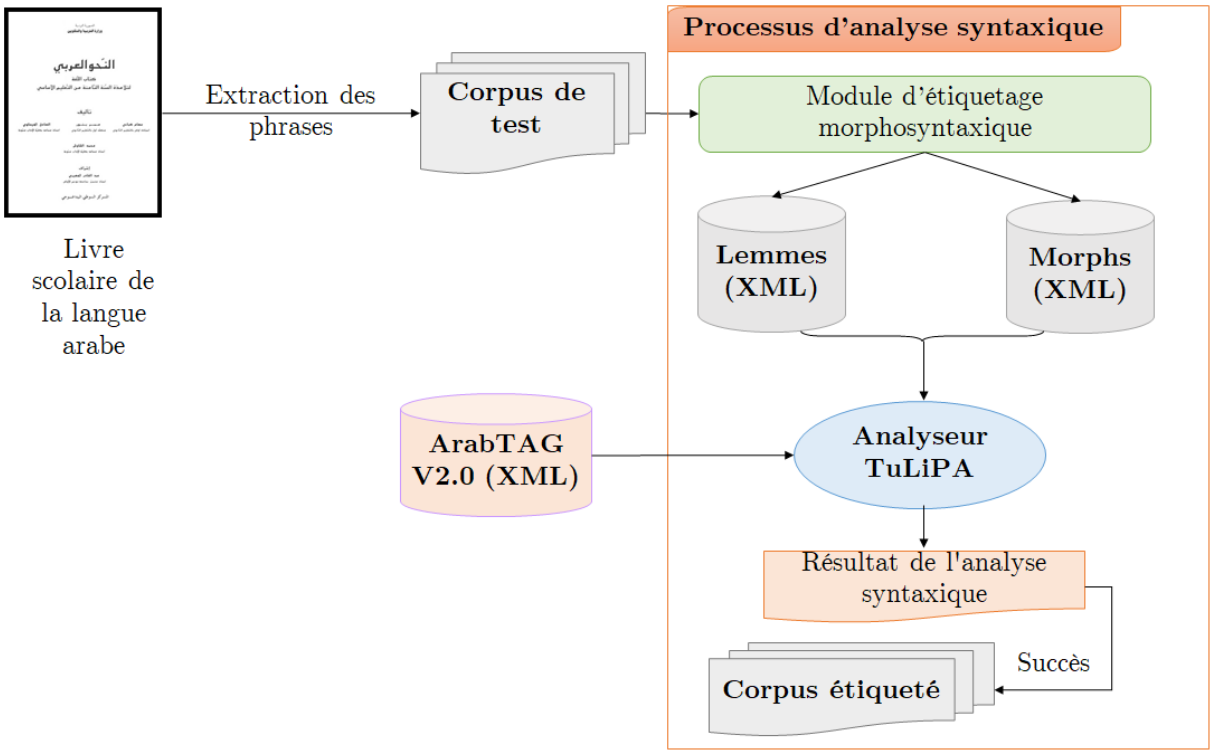


FIGURE 5.2: Processus d'analyse syntaxique

Nous allons à présent détailler chaque étape que nous venons de citer.

5.3.1.2.a Étiquetage morphosyntaxique

L'outil que nous avons développé, nous a permis de faciliter la phase de l'étiquetage morpho-syntaxique. Chaque mot de la phrase découpée se voit attribuer manuellement ses traits morphosyntaxique (et éventuellement son cadre sémantique) correspondants.

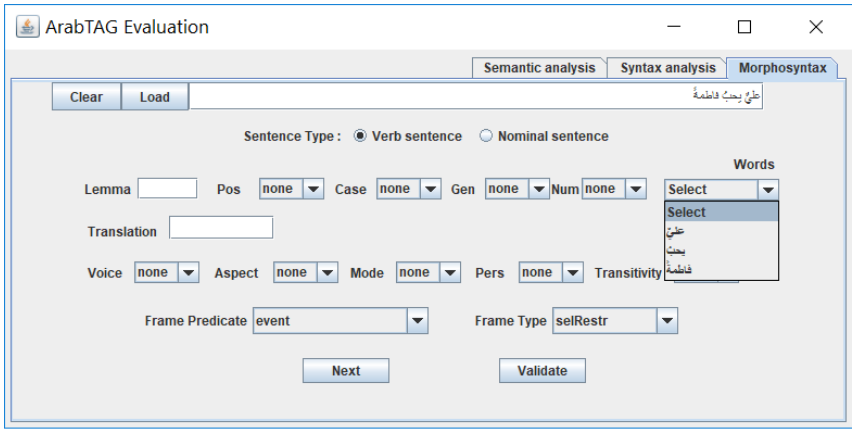


FIGURE 5.3: Interface graphique du module de l'étiquetage morphosyntaxique

Prenons l'exemple de la phrase "عليٌ يحبُ فاطمةً" (Ali aime Fatima). Cette phrase est découpée en trois mots : "عليٌ" (Ali), "يحبُ" (aime) et "فاطمةً" (Fatima). Ensuite, à chaque mot nous attribuons ses valeurs de traits. [4]
Cette phase d'étiquetage permet de créer notre base de lexique. Cette base est composée de deux fichiers aux formats biens définis. Le premier fichier contient le lexique morphologique tandis que le deuxième contient la base des lemmes.
La Figure suivante illustre un extrait du lexique morphologique de la phrase "عليٌ يحبُ فاطمةً" (Ali aime Fatima) suite à l'étiquetage morphosyntaxique.

علىٌ	علي	[cas=nom; pos=sn; gen=m]
فاطمةً	فاطمة	[cas=acc; pos=sn; gen=f]
يحبُ	أحب	[pers=3; num=sg; pos=v; gen=m]

FIGURE 5.4: Extrait du lexique morphologique de la phrase (Ali aime Fatima)

Chaque ligne du lexique morphologique correspond à une entrée morphologique. Cette dernière est constituée de trois champs : la forme fléchie, son lemme et enfin l'ensemble des traits qui lui sont associés, compatibles avec les traits spécifiés dans ArabTAG V2.0. La base de lemmes est analogue au lexique morphologique. Un extrait de cette base est illustré par la figure 5.5.

4. Nous avons utilisé Elixir-Fm pour vérifier les valeurs des traits attribuées.

%Ali	%Fatima	%aimer
*ENTRY: علي	*ENTRY: فاطمة	*ENTRY: أحب
*CAT: sn	*CAT: sn	*CAT: v
*SEM: *ACC: 1	*SEM: *ACC: 1	*SEM: *ACC: 1
*FAM: NomPropre	*FAM: NomPropre	*FAM: TransitifActif
*FILTERS: []	*FILTERS: []	*FILTERS: []
*EX: {}	*EX: {}	*EX: {}
*EQUATIONS:	*EQUATIONS:	*EQUATIONS:
*COANCHORS:	*COANCHORS:	*COANCHORS:

FIGURE 5.5: Extrait de la base des lemmes de la phrase (Ali aime Fatima)

À chaque entrée est associée son lemme, sa catégorie, sa famille d'arbres (NomPropre et TransitifActif) définie dans ArabTAG V2.0 et son cadre sémantique (nous reviendrons sur point dans la section 2.2.2.1 de ce chapitre). Eventuellement, nous pouvons aussi spécifier d'autres informations telles que le co-ancre. Par exemple, cette spécification nous a permis de gérer les verbes qui nécessitent un complément d'objet indirect. Le co-ancre détermine quelle est la préposition (du complément d'objet indirect) associable au verbe et intervient ainsi dans le processus d'analyse.
À partir de ces descriptions, des fichiers au format XML sont générés automatiquement en utilisant un compilateur spécifique appelé "lexconverter".

5.3.1.2.b Analyse syntaxique

L'analyse syntaxique est effectuée par l'analyseur TuLiPA [Parmentier et al., 2008]. Ce dernier reçoit en entrée : la phrase découpée en mots, la base des lemmes, la base des morphs et la grammaire ArabTAG V2.0. Ensuite, la lexicalisation (l'opération de l'ancrage lexical) est réalisée comme suit :

1. Chaque mot découpé de la phrase à analyser est associé à son lexique morphologique.
2. L'analyseur récupère les lemmes correspondants du lexique morphologique de la phrase.
3. Pour chaque lemme et sa forme fléchie, l'analyseur va tenter de leur associer les schèmes des familles d'arbres correspondantes de la grammaire. Cette correspondance revient à unir les structures des traits du nœud ancré avec ceux de la forme fléchie (les traits morphosyntaxiques).
4. Si l'unification des traits réussit, l'analyseur va ajouter la forme fléchie du mot (avec ses traits morphosyntaxiques) en dessous du nœud ancre
5. Enfin, le résultat final (l'arbre dérivé de la phrase et son arbre de dérivation) de l'analyse est généré.

Reprenons l'exemple de la phrase "علِيٌ يحبُ فاطمةً" (Ali aime Fatima). TuLiPA récupère les couples de (lemme, forme fléchie) suivant : (علي,علِيٌ) pour Ali , (أَحَبَّ, يحبُ) pour aime et (فاطمة,فاطمةً) pour Fatima. Chaque lemme de ces couples sera associé à tous les arbres

de la famille dont il est rattaché. Pour le verbe aimer, il sera associé à tous les arbres de la famille TransitifActif, tandis que Ali et Fatima seront associés à ceux de NomPropre. Une fois l'ancrage et l'instanciation de traits de ces unités lexicales sont réalisés avec succès, nous aurons le résultat illustré par la figure 5.6.

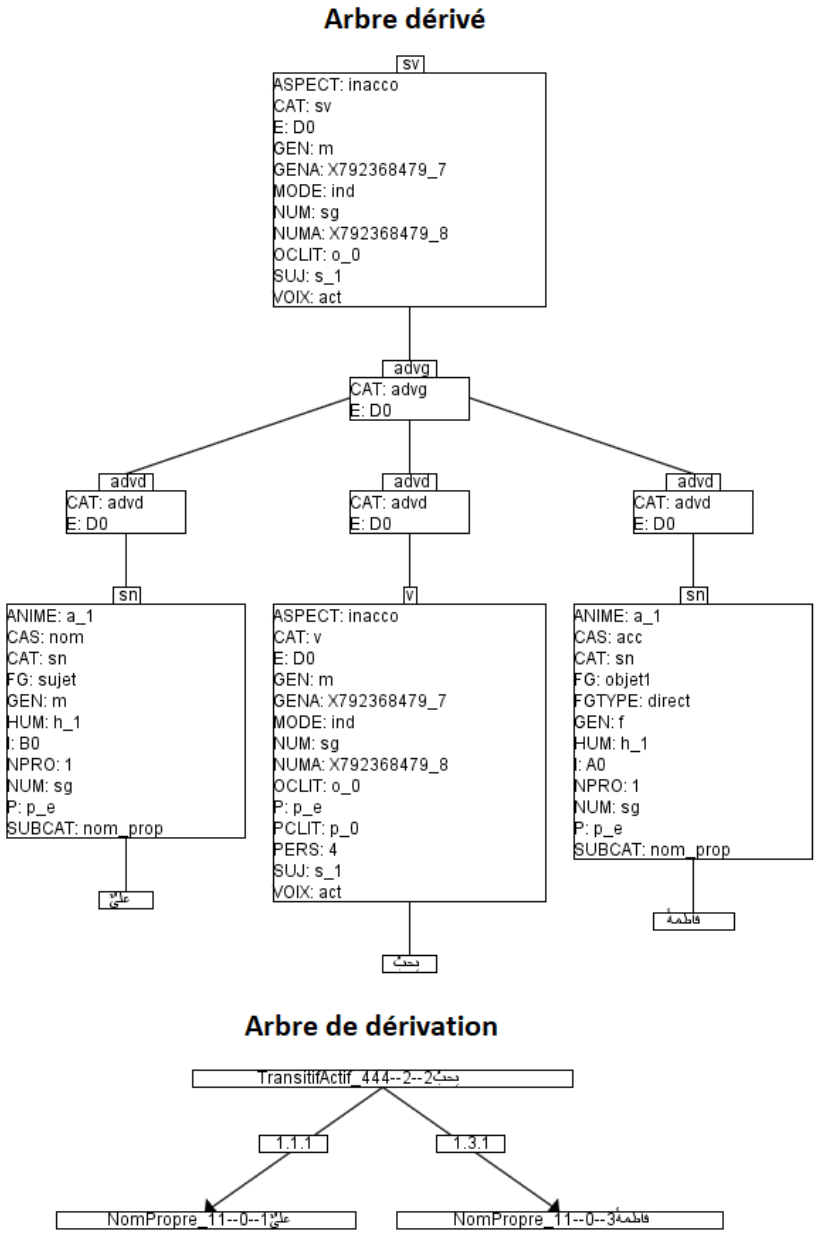


FIGURE 5.6: Résultat de l'analyse syntaxique de la phrase (Ali aime Fatima)

Dans certains cas, nous pouvons avoir plus d'un seul arbre syntaxique résultat. Dans ce cas nous parlons d'ambiguïté d'interprétation syntaxique de la phrase.

5.3.1.2.c Résultat de l'analyse syntaxique

Après avoir obtenu l'arbre syntaxique correspondant à la phrase analysée, ce résultat est stocké dans un nouveau fichier (sous format xml). En effet, cette étape d'évaluation ne permet pas seulement de vérifier la qualité de notre grammaire, mais en plus de construire de nouvelles ressources lexicales pour l'arabe que nous voulons mettre à disposition pour d'autres travaux de recherche.

5.3.1.3 Résultat de l'analyse syntaxique

Comme nous l'avons précisé précédemment, nous avons utilisé deux corpus pour l'évaluation syntaxique :

— Un premier corpus constitué de 1000 phrases grammaticalement correctes.

— Un deuxième corpus constitué de 250 phrases agrammaticales.

Les résultats d'analyse du premier corpus sont regroupés dans le tableau 5.1.

Critères d'évaluation		**Mesures**
Précision		82,33%
Rappel		88,10%
F1-mesure		85,11%
Nombre d'analyses réussies et taux de succès	Phrases verbales	611
	Phrases nominales	270
	Total	881 (88,1%)
Nombre de phrases non analysées et taux d'échec	Phrases verbales	39
	Phrases nominales	80
	Total	119 (11,9%)
Nombre de phrases ayant plus qu'une analyse résultante	Phrases verbales	39
	Phrases nominales	67
	Total	106 (12,03%)

TABLE 5.1: Résultats de l'analyse syntaxique du corpus de test

Comme l'illustre le tableau 5.1, nous avons réussi à analyser correctement 88,1 % des phrases du corpus de test. En d'autres termes, l'analyseur a réussi à trouver au moins un arbre syntaxique correspondant à la phrase. Ce résultat, nous l'avons validé par apport à une référence manuelle. Les taux de précision et de rappel mesurés attestent de la faiblesse du bruit et silence générés durant l'analyse. Par ailleurs, le nombre de phrases ambiguës s'élève à 106. Ce résultat s'explique par l'ambigüité d'interprétations syntaxiques possibles de certaines phrases. Par conséquence, l'analyseur syntaxique a fourni tous les arbres résultats correspondants aux interprétations syntaxiques possibles de ces phrases.

Considérons l'exemple de la phrase "ودع المُدَرِّس عدد كبير من طلبة المدرسة و مدرسيها" (Un grand nombre d'élèves de l'école et ses enseignants ont salué l'enseignant). L'analyse syntaxique de cette phrase a donné comme résultat deux arbres syntaxiques distincts illustré par la figure 5.7.

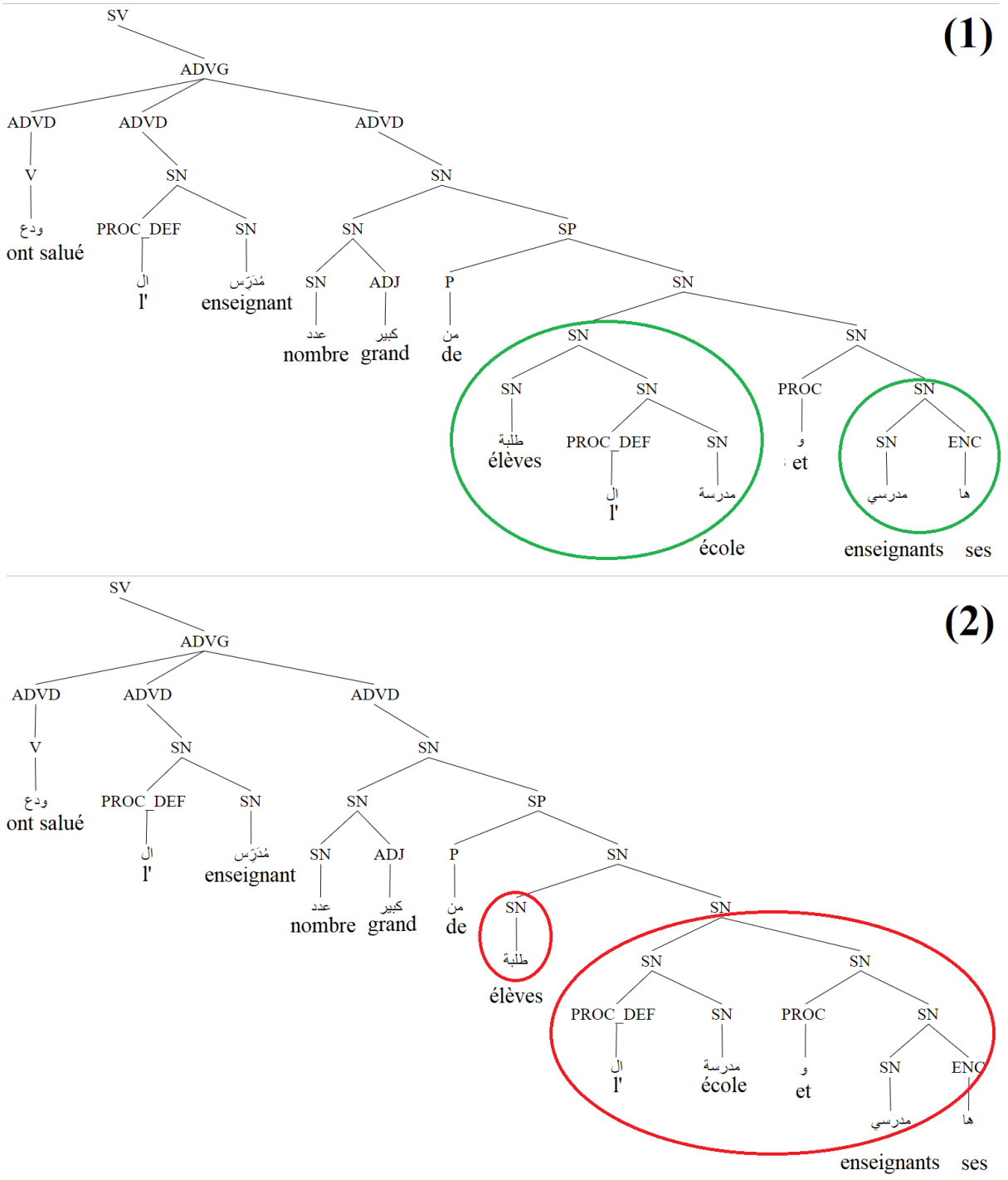


FIGURE 5.7: Résultat de l'analyse syntaxique de la phrase (Un grand nombre d'élèves de l'école et ses enseignants ont salué l'enseignant)

Ces deux interprétations syntaxiques s'expliquent par l'ambiguïté au niveau des syntagmes composants la structure "طلبة المدرسة و مدرسيها" (élèves de l'école et ses enseignants). Dans le premier modèle, cette structure est un syntagme de conjonction composé au moyen de deux syntagmes d'annexion : "طلبة المدرسة" (élèves de l'école) et "مدرسيها" (ses enseignants). Dans le second modèle, nous remarquons le changement de la structure. Cette dernière est un syntagme d'annexion composé d'une conjonction entre : "المدرسة" (l'école) et "مدرسيها" (ses enseignants).
Bien que dans cet exemple les deux interprétations sémantiques soient plausibles, la première reste la plus adéquate. Dans la suite de ce chapitre, nous allons voir comment dans certains cas nous pouvons résoudre l'ambiguïté d'interprétation syntaxique en faisant intervenir l'information sémantique.
Par ailleurs, nous avons obtenu un taux d'échec de 11,9%. Ce taux obtenu représente le nombre de 80 phrases nominales et 39 phrases verbales pour lesquelles l'analyseur a été silencieux. Quelques cas récurrents de cet échec sont analysés dans le tableau 5.2.

N°	**Phrase**	**Type de phrase**	**Cas d'échec**	**Solution**
1	هَذِه صُفُوفٌ مِنَ البُيوت والقُصورِ (Ce sont des rangées de maisons et de palais)	Nominale	L'accord entre le pronom démonstratif et le syntagme nominale n'est pas pris en compte par les traits de la grammaire.	Rajouter le trait d'accord manquant
2	مَاأقلَّ حظِ يبَيَنَ الرّيَاحِينِ (quel manque de chance parmi ces basilics)	Nominale	Structure d'exclamation non définie par la grammaire.	Rajouter les descriptions correspondantes aux modèles manquants dans la méta-grammaire
3	مُنطَلقُ القافلةِ بعدَ ساعةٍ (Départ du convoi après une heure)	Nominale	Le thème est non défini, mais puisque c'est un syntagme d'annexion la structure est correcte.	Rajouter la catégorie du syntagme d'annexion (non défini) dans la description du thème
4	السُّرُرُ الخَشبيَّةُ المَنقوشَةُ كَانَت مِنْ فَنّ مغربيٍّ (Les lits en bois sculptés étaient de l'art séduisant)	Nominale avec verbe d'exsistance	Le verbe d'existence appartient au propos et non pas au thème de la phrase	Ajouter la possibilité d'avoir un verbe d'existence qui commence le propos

5	أنا الذي عرف ما في الإنسان لكثرة ملاحظاتي له (Moi qui savais ce qu'il y a chez l'homme à cause de mes multitude observations de lui)	Nominale	Une particule interrogative qui commence le syntagme d'un complément d'objet	Rajouter les descriptions des fragments de ce modèle manquant dans la méta-grammaire
6	تَماوتَ الثعلَبُ لِيَنْجُوَ مِنَ الصّياد (le renard a feint sa mort pour échapper au chasseur)	Verbale	Complément de raison avec une particule suivie d'un syntagme verbale non reconnue	Rajouter les descriptions des fragments de ce modèle manquant dans la méta-grammaire
7	جاءتْ أختِي و جاءَ أخوايَ (Ma soeur est venue et mes deux frères sont venus)	Verbale	Conjonction entre deux syntagmes verbaux	Ajouter un modèle permettant la conjonction de deux syntagmes verbaux
8	لاأذكُرُ أنّي حزِنتُ لِموتِ خَالي (Je ne me souviens pas d'avoir été triste à cause de la mort de mon oncle)	Verbale	Le complément d'objet est composé d'une structure nominale avec un verbe d'existence	Autoriser la composition de l'ensemble des fragments permettant la définition de cette structure d'une phrase verbale

TABLE 5.2: Quelques exemples d'échecs de l'analyse syntaxique

Comme le montre le tableau 5.2, la majorité des erreurs sont dues à la non-reconnaissance de certaines structures de phrases plus complexes que celles décrites dans notre grammaire (exemples : 2, 3, 4, 5 et 6). Nous avons aussi souligné quelques exemples non reconnus en raison des contraintes imposées par les structures de traits ce qui a engendré la sous-génération de quelques modèles (exemples : 1, 7 et 8). Néanmoins, il ne nous est impossible de remédier à ces deux points. Comme l'illustre le tableau 5.2, nous avons proposé une solution réalisable pour chaque exemple non analysé. En effet, l'extensibilité de notre méta-grammaire facilite sa mise à jour et favorise la modification et l'ajout de telles descriptions permettant ainsi d'étendre de plus en plus sa couverture des structures syntaxiques de l'arabe.
La deuxième phase d'évaluation syntaxique, consiste à tester le comportement d'Arab-TAG V2.0 lors de l'analyse des phrases agrammaticales. Nous avons analysé les 250 phrases agrammaticales que nous avons construites manuellement à partir de notre corpus de test (1000 phrases). Les résultats sont fournis dans le tableau 5.3.

Critères d'évaluation		**Mesures**
Précision		100%
Rappel		86,8%
F1-mesure		92,93%
Nombre d'analyses réussies et taux de succès	Phrases verbales	18
	Phrases nominales	15
	Total	33 (13,02%)
Nombre de phrases non analysées et taux d'échec	Phrases verbales	149
	Phrases nominales	68
	Total	217 (86,8%)

TABLE 5.3: Résultats de l'analyse syntaxique des 250 phrases agrammaticales

Comme indiqué dans le tableau 5.3, nous avons mesuré un taux de précision de 100% indiquant l'absence de bruit. Autrement dit, aucune des phrases n'est déclarée à tort par l'analyseur comme étant grammaticalement incorrecte. En revanche, nous avons observé un taux de silence de 13,02%. Ce nombre représente 18 phrases verbales et 15 phrases nominales. Nous avons étudié ces phrases et défini les raisons de ce résultat non souhaitable. Le tableau 5.4 expose quelques-unes de ces phrases.

N°	**Phrase**	**Type de phrase**	**Type de l'erreur grammaticale**
1	عادَت جاريًا إلى الشّجيرَةِ (elle est retournée à l'arbrisseau en courant)	Verbale	Accord du genre entre le verbe (courir) congé au féminin singulier et le dérivé (en courant) du syntagme de mode qui est au masculin singulier
2	شَكَيتُ حَمزَةُ هُمُوم اَقَلبِهِ (J'ai exprimé Hamza les soucis de son cœur)	Verbale	Accord du nombre entre le verbe et le sujet. Le sujet et le verbe sont au singulier mais le verbe est conjugué à la première personne alors qu'il devrait être à la 3ième personne pour s'accorder avec son sujet "Hamza"
3	أيْن مِنَ الدِّراسةِ (où est ce que des études)	Nominale	Les structures des phrases interrogatives sont très variées selon la particule d'interrogation. Cette phrase nominale est une phrase interrogative composée d'une particule interrogative suivie par un syntagme prépositionnel. Cette structure peut être valable pour une autre particule mais pas pour le cas de "où", elle est donc incomplète.

4	علم شجرة (savoir arbre)	Nominale	Le thème de la phrase aurait dû être défini pour qu'elle soit syntaxiquement correcte "العلم شجرة" (le savoir est un abre)
5	هِي أبلَهُ الطبْعِ (elle est stupide)	Nominale	L'accord en genre entre le thème (pronom féminin singulier "هِي" elle) et le propos qui est au singulier masculin
6	إن سريع الحركة (il est rapide)	Nominale	La phrase nominale commence par un verbe de certitude suivi par son propos sans le thème. Ce dernier n'est pas inclusif dans la phrase.

TABLE 5.4: Quelques exemples agrammaticaux analysés

Comme indiqué dans le tableau 5.4, l'analyse de ces exemples agrammaticaux s'explique par l'insuffisance des contraintes imposées au sein des descriptions de leurs arbres élémentaires. Par exemple, pour la phrase (1) et (2) il faudra s'assurer du bon accord en genre, respectivement en nombre, entre le verbe et le sujet. Il est donc nécessaire de rajouter des conditions sur les traits genre et personne pour interdire ce genre d'accord. Il en va de même pour la phrase nominale des exemples (4) et (5).
Cependant, nous avons noté que se sera plus pertinent d'affiner certaines descriptions de structures telles que les phrases interrogatives (exemple 3) ou encore les phrases nominales commençants par un verbe de certitude (exemple 6).

5.3.2 Évaluation de l'analyse sémantique

Nous entamons cette section par une présentation du corpus de test choisi pour évaluer la couverture sémantique de notre grammaire. Par la suite, nous retraçons les étapes de cette évaluation avant de terminer par une discussion des résultats mesurés.

5.3.2.1 Corpus de test

Afin de pouvoir comparer nos résultats d'analyse sémantique avec des exemples de phrases étiquetées sémantiquement, nous avons choisi de construire un corpus composé de phrases verbales extraites de la ressource lexicale ArabicVerbNet. En effet, pour chaque type de cadre sémantique défini dans ArabicVerbNet, au moins un exemple de phrase étiquetée par des rôles sémantiques lui est attribué. Nous avons voulu comparer le résultat de notre analyse syntaxico-sémantique avec l'étiquetage fait au sien d'ArabicVerbNet et aussi s'assurer que ArabTAG V2.0 peut couvrir ces différents types de cadres sémantiques. Nous avons donc extrait [5] une phrase exemple de chaque cadre sémantique décrit dans ArabicVerbNet. A l'issue de cette extraction, nous avons obtenu un corpus de 460 phrases. Comme illustré par la figure 5.8, l'ensemble des 460 de phrases est composé de 30 phrases

5. L'extraction est réalisée aléatoirement grâce à un programme informatique que nous avons développé.

à verbes intransitifs (nécessitent un seul argument), 201 phrases à verbes transitifs (nécessitent deux arguments) et 229 phrases à verbes ditransitifs (nécessitent trois arguments). La longueur de ces phrases varie entre 2 et 14 mots.

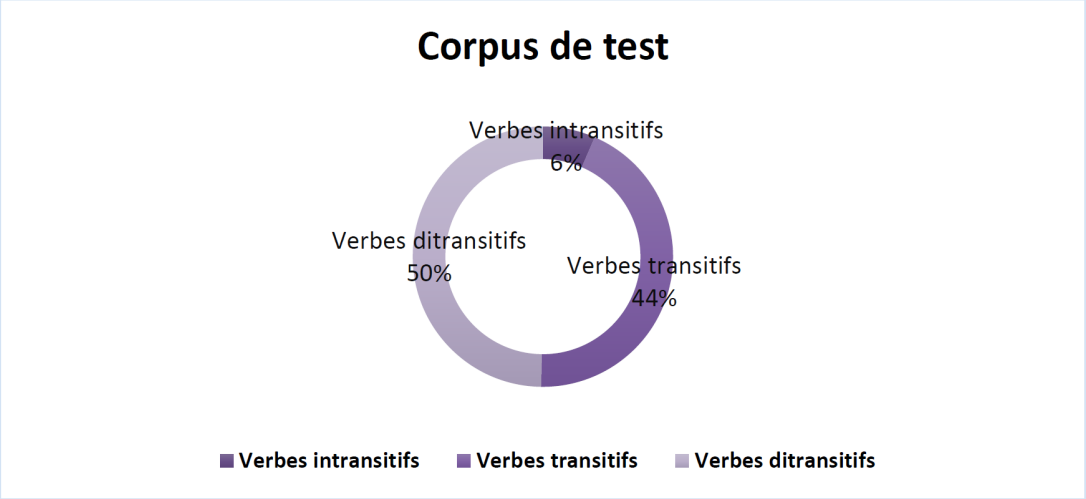


FIGURE 5.8: Répartition des phrases du corpus de test selon la transitivité du verbe

5.3.2.2 Processus de l'évaluation sémantique

Le processus d'analyse sémantique (plus précisément syntaxico-sémantique) suit les mêmes étapes de l'analyse syntaxique. En effet, les phrases du corpus de test vont subir l'étape de l'étiquetage morphosyntaxique et sémantique ainsi qu'une analyse syntaxico-sémantique. A la fin de cette dernière nous comparons le(s) représentation(s) sémantique(s) résultante(s) avec celle d'ArabicVerbNet. La figure 5.9 illustre le processus d'analyse.

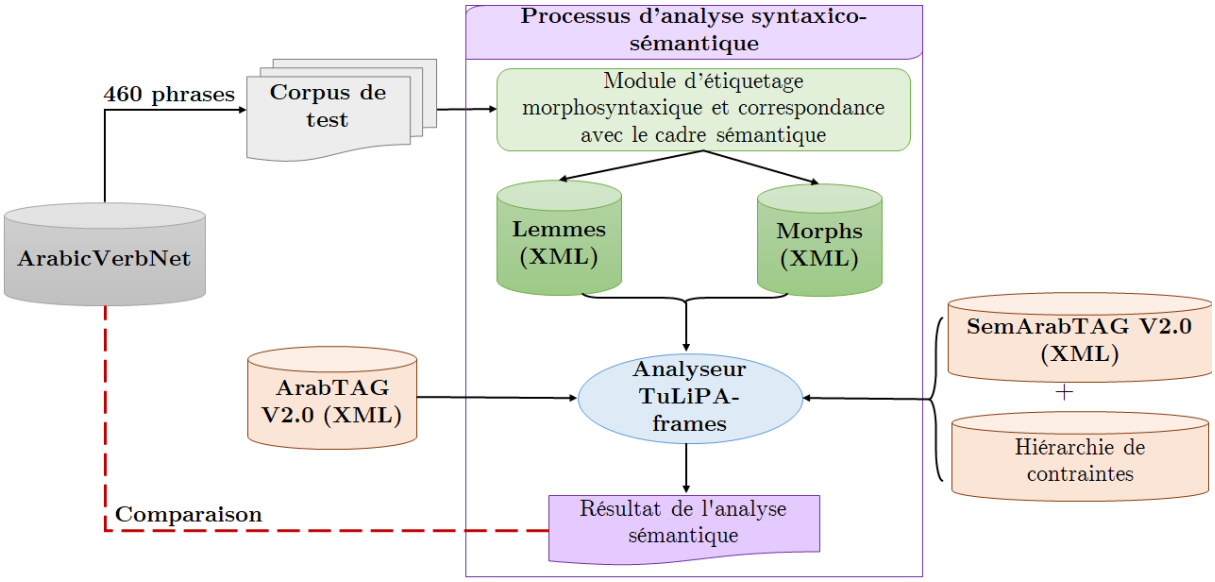


FIGURE 5.9: Processus d'analyse syntaxico-sémantique

Tout d'abord, les mots de la phrase en entrée subissent un étiquetage morphosyntaxique accompagné d'une correspondance sémantique avec les familles de cadres sémantiques définies dans la méta-grammaire. À la suite de cette étape, les bases de lemmes et de morphes saisies sont générées. Ensuite, l'analyse syntaxico-sémantique est réalisée au moyen d'une nouvelle version de l'analyseur TuLiPA à savoir TuLiPA-frames [Arps and Petitjean, 2018]. Enfin, le résultat de l'analyse sémantique est comparé à la représentation sémantique de la phrase au sein d'ArabicVerbNet.

5.3.2.2.a Correspondance sémantique

Durant la première étape de l'étiquetage morphosyntaxique (voir section 2.1.2.1. de ce chapitre), la correspondance des unités lexicales de la phrase en entrée avec leurs informations sémantiques est assurée. Ces informations saisies au moyen de l'outil que nous avons mis en place, vont permettre la sélection et l'affectation de la famille de cadres sémantiques correspondantes.
En effet, lorsque le mot saisi est un verbe, l'outil nous permet de préciser l'évènement (par exemple : activité, émotion, transformation, etc.) lié à ce verbe en plus de ses informations morphosyntaxiques. Une fois notre saisie est validée, l'outil dans un premier temps, récupère l'information sur la transitivité du verbe qui permet de lui affecter sa famille syntaxique ainsi que sa famille de cadre sémantique. Ensuite, il se connecte avec la base des verbes extraite d'ArabicVerbNet afin de récupérer les rôles sémantiques (éventuellement leurs contraintes) correspondants au verbe saisi. Ces rôles (et éventuellement contraintes) sont passés en paramètre dans la famille de cadre sémantique du verbe.
Lorsque le mot saisi n'est pas un verbe, l'outil met à disposition une liste qui regroupe tous les types de cadre définis dans notre hiérarchie de contraintes (voir section 2.1.2. du chapitre 4). Une fois la saisie des informations morphosyntaxiques ainsi que le type du cadre est validée, l'outil attribue au mot sa famille syntaxique et sa famille de cadre sémantique. Cette dernière reçoit en paramètre le type du cadre.

%Ali
*ENTRY: علي
*CAT: sn
*SEM: FrameNomPropre [**Lem**= علي , **FEType**= human]
*ACC: 1
*FAM: NomPropre
*FILTERS: []
*EX: {}
*EQUATIONS:
*COANCHORS:

%Fatima
*ENTRY: فاطمة
*CAT: sn
*SEM: FrameNomPropre [**Lem**= فاطمة ,**FEType**= human]
*ACC: 1
*FAM: NomPropre
*FILTERS: []
*EX: {}
*EQUATIONS:
*COANCHORS:

%aimer
*ENTRY: أحب
*CAT: v
*SEM: FrameTransitif [**NewFrameType**= desire, **First**= experiencer, **Constraint1**= animate, **Second**= patient, **Constraint2**= animate]
*ACC: 1
*FAM: TransitifActif
*FILTERS: []
*EX: {}
*EQUATIONS:
*COANCHORS:

FIGURE 5.10: Extrait de la base des lemmes des mots (Ali) (Fatima) et (aimer)

Reprenons l'exemple de la phrase "علي يحبُ فاطمةً" (Ali aime Fatima) découpée en trois mots : "علي" (Ali), "يحبُ" (aime) et "فاطمةً" (Fatima). La nouvelle base de lemmes, suite à l'étiquetage morphosyntaxique et la correspondance avec les cadres sémantiques, est illustrée par la figure 5.10.
Le champ "SEM" indique la famille de cadre sémantique affectée au mot. Dans cet exemple, les noms "علي" (Ali) "فاطمة" (Fatima) sont associés à la famille de cadre sémantique des noms propres. Cette dernière permet d'attribuer au cadre sémantique son type (humain) et la valeur du lemme du nom propre (علي / Ali ou فاطمة /Fatima). Le verbe "أَحَبَّ" (aimer) est associé à sa famille de cadre sémantique de verbe transitif. Cette classe spécifie l'événement du verbe aimer (desire), ses deux rôles sémantiques (experiencer et patient) ainsi que leurs contraintes (animate).

5.3.2.2.b Analyse syntaxico-sémantique

L'analyse syntaxico-sémantique est effectuée au moyen de l'analyseur TuLiPA-frames [Arps and Petitjean, 2018]. Ce dernier est une extension de TuLiPA [Parmentier et al., 2008] qui permet de fournir, en plus de l'arbre syntaxique résultat, son ou ses cadre(s) sémantique(s) correspondant(s).

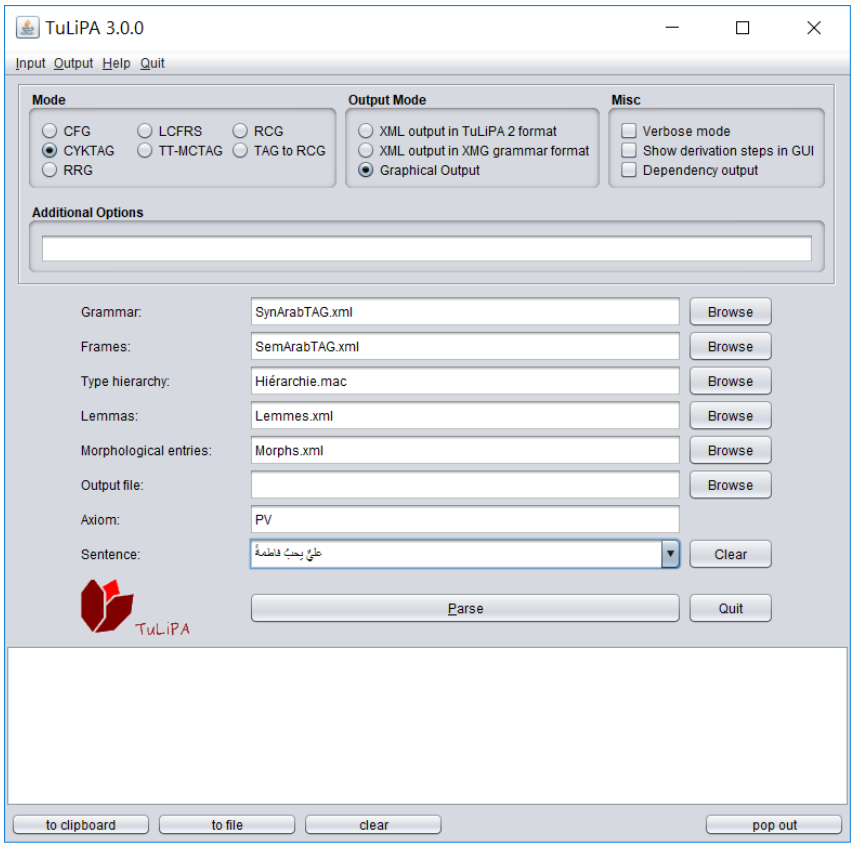


FIGURE 5.11: L'interface d'utilisation de TuLiPA-frames

TuLiPA-frames reçoit en entrée (figure 5.11) : la description de la syntaxe (SynArabTAG), celle de la dimension sémantique (SemArabTAG), la hiérarchie des contraintes de rôles sémantiques et de type de cadre sémantiques élémentaires, la base des lemmes, la base des morphs et enfin la phrase découpée.
Si l'unification des cadres sémantiques est un succès, l'analyseur fournit un (ou plusieurs) résultat(s) composé de l'arbre dérivé de la phrase, son arbre de dérivation et son cadre sémantique final.
Reprenons l'exemple de la phrase "عليٌ يحبُ فاطمةً" (Ali aime Fatima). Le cadre sémantique du prédicat "أَحبَّ" (aimer) possède deux rôles sémantiques de type "Patient" : le premier est un "Expriencer" qui est un patient conscient de subir l'événement (désirer) tandis que le second est un "Patient" qui subit cet événement. Ces deux rôles doivent être animés. Lors de l'unification, le type des cadres sémantiques élémentaires, de "علي" (Ali) "فاطمة" (Fatima), doivent être compatibles avec la contrainte "animé" pour que l'unification n'échoue pas. La hiérarchie de contraintes que nous avons implémentée autorise l'unification d'un humain avec un animé. Ainsi, le cadre sémantique final associé à l'arbre syntaxique résultat est représenté comme suit :

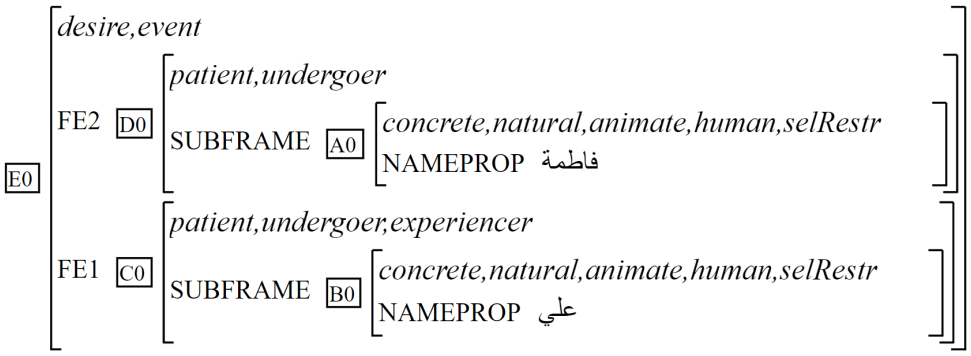


FIGURE 5.12: Représentation sémantique de la phrase (Ali aime Fatima)

Cependant, en voulant analyser la phrase "يحبُ الكتابُ فاطمةً" (le livre aime Fatima) qui est syntaxiquement correcte, l'analyse syntaxico-sémantique a échoué. Cet échec s'est produit car le sujet "الكتابُ" (livre) est un objet non animé. En d'autres termes, l'unification entre le cadre sémantique du prédicat, dont l'experiencer doit être animé, ne va pas s'unifier avec le cadre sémantique élémentaire du livre qui est non animé.
En plus des contraintes sur les types des cadres sémantiques (des rôles), nous avons considéré un autre moyen de restriction afin d'optimiser l'analyse sémantique. Ce moyen consiste à faire intervenir le type de la préposition des syntagmes prépositionnels. En effet, certains rôles sémantiques ont tendance à apparaître comme des syntagmes prépositionnels. Dans ce cas, la préposition peut aider à déterminer le sens de ce syntagme et ainsi intervenir pour restreindre le choix du cadre correspondant.

Considérons l'exemple de la phrase "نبحَ الكلبُ منَ الخوفِ" (le chien aboie de peur).
Le verbe (transitif) prédicat des deux phrases est "نبح" (aboyer). Nous avons remarqué que lors de la mise en correspondance sémantique, son lemme est affecté à trois combinaisons possibles de cadres sémantiques :

(a) Agent+ prep (على) + Recipient : la préposition "على" (sur)indique que le rôle sémantique de l'objet est Recipient.

(b) Agent+ prep (من) + Cause : la préposition "من" (de) exige que le rôle sémantique de l'objet soit Cause.

(c) Location+ prep (بِ) + Agent : la préposition "بِ" (avec) indique que le rôle sémantique de l'objet est Agent.

Puisque les particules sont définies comme des co-ancres dans ArabTAG V2.0, l'analyse syntaxico-sémantique tiendra compte de la contrainte sur la préposition pour réaliser correctement l'unification. Finalement, nous avons obtenu une seule correspondance sémantique :

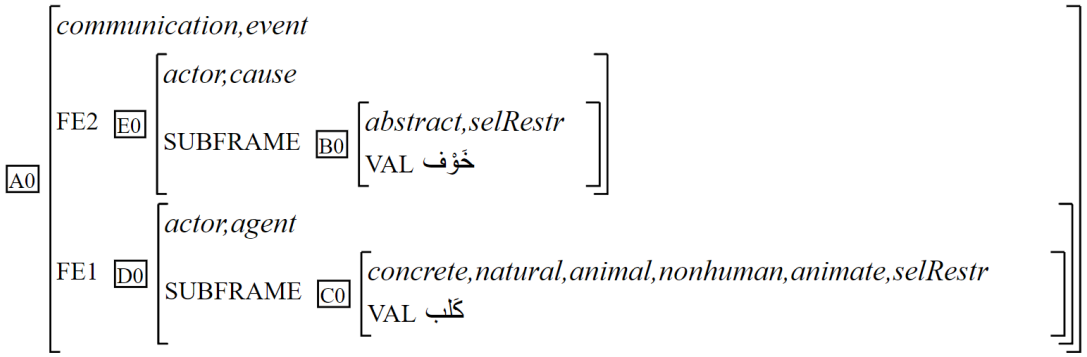


FIGURE 5.13: Cadre sémantique résultat de la phrase (Le chien aboie de peur)

Comme illustré par la figure 5.13, le cadre résultant respecte bien la structure du cadre sémantique (b) puisqu'il est composé des deux acteurs : l' Agent "كلب" (chien) qui est l'instigateur de l'évènement et la Cause "خوف" (peur) initiatrice de son aboiement.

5.3.2.3 Résultat de l'analyse sémantique

Le corpus d'évaluation consiste à un ensemble de 460 phrases verbales extraites des exemples d'ArabicVerbNet. Les résultats d'analyse sémantique de ces phrases sont présentés dans le tableau 5.5.
Comme l'illustre le tableau 5.5 nous avons réussi à analyser correctement 95,43% des phrases du corpus. Parmi les phrases correctement analysées 33 phrases possèdent plus qu'une représentation sémantique. La cause d'un tel résultat peut s'expliquer par le manque de restriction au niveau de la classe du verbe dans ArabicVerbNet : un verbe se voit attribuer plusieurs cadres sémantiques sans possibilité de filtrage. Ceci engendre en plus du résultat correcte un résultat susceptible d'être incorrecte.

Critères d'évaluation	**Mesures**
Précision	95,63%
Rappel	95,43%
F1-mesure	95,52%
Nombre d'analyse réussie et taux de succès	439 (95,43%)
Nombre de phrases non analysées et taux d'échec	21 (4,56%)
Nombre de phrases ayant plus qu'une analyse résultante	33

TABLE 5.5: Résultats de l'analyse sémantique

A titre d'exemple, l'analyse de la phrase "وَاظَبَ المُصَلِّي عَلَى المَسْجِد" (le fidèle persévère à aller[6] à la mosquée) dont le prédicat est le verbe "وَاظَبَ" (persévérer). Lors du mapping avec ArabicVerbNet, le prédicat a été associé à plusieurs cadres sémantiques possibles :[7]

(a) Agent (animate)+Predicate : L'agent doit être compatible avec animé.

(b) Agent (organization)+Predicate : L'agent doit être une organisation.

(c) Theme+Time (timeC) : Le deuxième rôle Time doit être de type temps.

(d) Theme+Cause : Aucune restriction sur le thème et la cause.

L'analyse syntaxico-sémantique a permis d'éliminer deux cadres sémantiques qui ne respectent pas les contraintes de la phrase. Cependant nous avons les deux résultats suivants : Le premier illustré par la figure 5.14 correspond à l'unification des cadres élémentaires de "المُصَلِّي" (le fidèle) et "عَلَى المَسْجِد" (à la mosquée) avec le cadre a) du prédicat, à savoir <Agent (animate)+Predicate>.

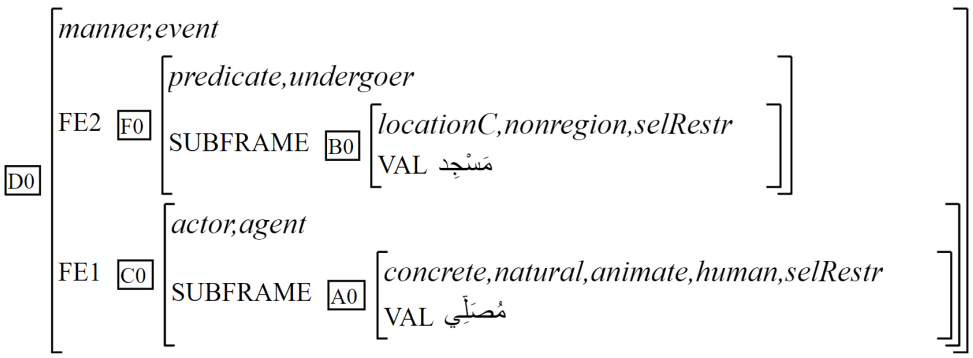


FIGURE 5.14: Premier cadre sémantique résultat de l'analyse de la phrase (Le fidèle persévère à aller à la mosquée)

6. La traduction mot par mot est : persévère le fidèle à la mosqué
7. La classe verbale n'exige pas de contrainte sur le type de préposition.

Tandis que le deuxième résultat représenté par la figure 5.15, correspond à l'unification des mêmes cadres élémentaires avec le cadre sémantique d) du prédicat : Theme+Cause. Après avoir effectué une comparaison entre ces deux résultats et l'étiquetage de l'exemple au sein d'ArabicVerbNet, nous avons pu conclure que c'est le premier cadre sémantique qui est correct. Mais le manque de restriction au niveau du deuxième cadre sémantique du prédicat, a autorisé l'unification et un résultat supplémentaire qui est erroné.

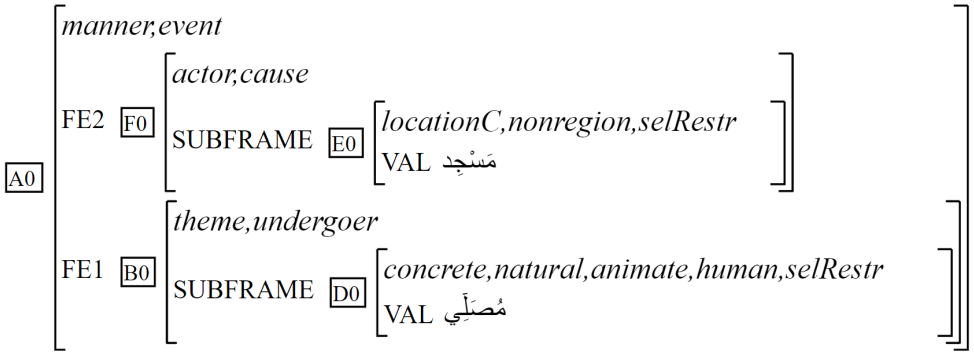


FIGURE 5.15: Deuxième cadre sémantique résultat de l'analyse de la phrase (Le fidèle persévère à aller à la mosquée)

Le taux d'échec mesuré durant notre analyse syntaxico-sémantique est de 4,56%. Dans certains cas, ceci est dû à l'échec de l'analyse au niveau syntaxique.

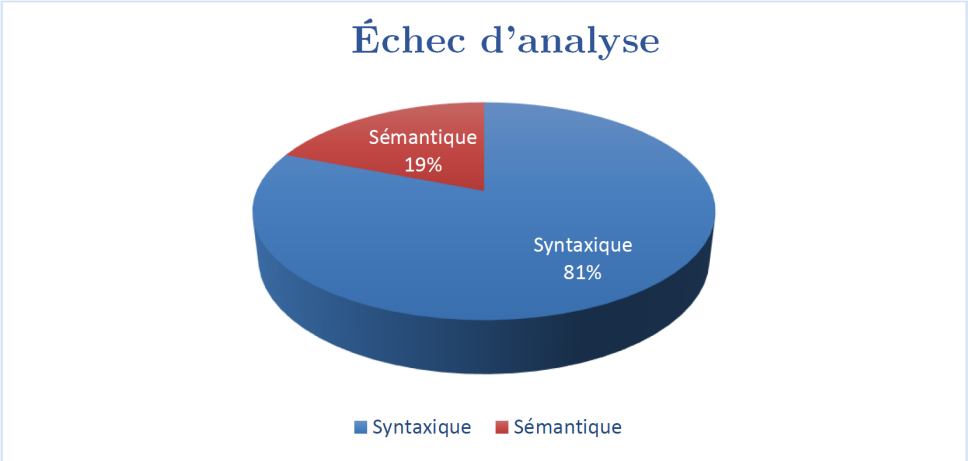


FIGURE 5.16: Répartition des cas d'échec de l'analyse syntaxico-sémantique

Comme l'illustre la figure 5.16, la majorité des cas (81%) d'échecs sont obtenus au niveau de l'analyse syntaxique. L'analyse syntaxique a échoué pour 17 phrases. Ce résultat s'explique par la non-couverture d'ArabTAG V2.0 de ces structures. En revanche, l'analyse sémantique a échoué pour 4 phrases seulement. Ce résultat s'explique par l'échec d'unification au niveau des cadres élémentaires définis pour certains types de syntagmes (par exemple approbatif et corroboratif). Afin de résoudre ce problème, nous envisageons d'affiner les descriptions que nous avons défini pour nos cadres sémantiques élémentaires et ainsi autoriser leur unification.

5.3.2.4 Analyse sémantique pour désambiguïser la représentation syntaxique

Dans certains cas, la sémantique permet de lever l'ambiguïté d'interprétation syntaxique dans une sorte de retour arrière. Nous avons testé notre démarche sur l'exemple de la phrase "أَكَلَ الضَيْفُ الكثيرَ من اللَحْم مع المُضِيف" (L'invité a mangé beaucoup de viande avec l'hôte). Cette phrase admet deux arbres syntaxiques possibles, illustrés par la figure 5.17.

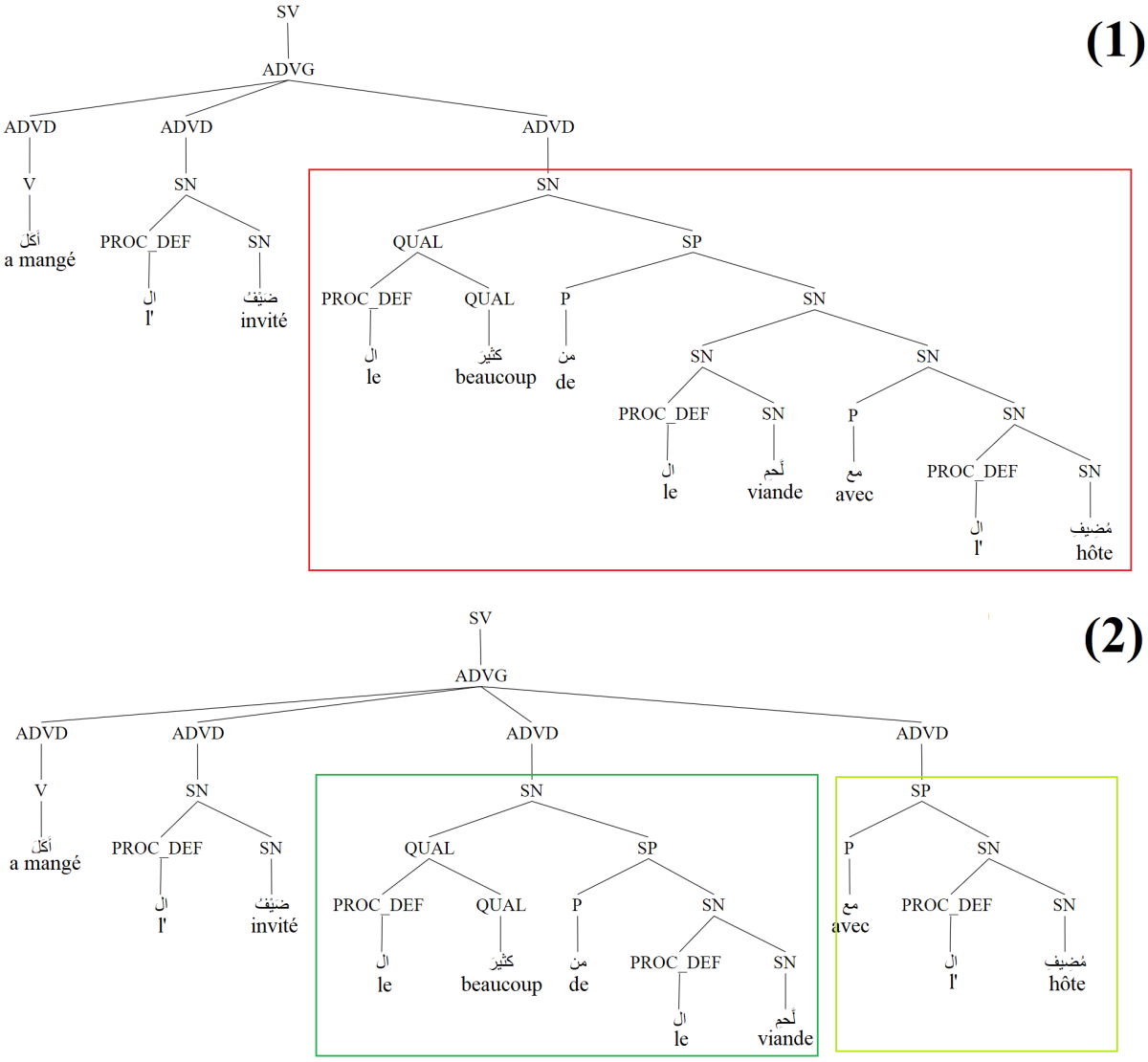


FIGURE 5.17: Représentations syntaxiques de la phrase (L'invité a mangé beaucoup de viande avec l'hôte)

L'ambiguïté de l'interprétation syntaxique est présente au niveau du syntagme : "الكثيرَ من اللَحْم مع المُضِيف" (beaucoup de viande avec l'hôte). La première représentation syntaxique est la suivante :

((الكثيرَ (من (اللَحْم مع المُضِيف))

((beaucoup (de (viande avec l'hôte))

Par conséquence, le sens de la phrase indique que l'invité est en train de manger l'hôte et beaucoup de viande. Cependant, cette signification est erronée puisque l'invité ne peut pas manger l'hôte.
Tandis que le second modèle syntaxique décrit une autre structure :

(مع المُضِيف) ((من اللَّحم) الكثيرَ)

(beaucoup (de viande)) (avec l'hôte)

Nous remarquons cette structure est séparée en deux syntagmes distincts. De ce fait la signification de la phrase change. Le sens dégagé de ce modèle indique que l'invité et l'hôte mangent ensemble beaucoup de viande, ce qui correspond à la bonne interprétation. Afin d'affirmer ces observations, nous avons procédé à une analyse syntaxico-sémantique de cet exemple. A l'issue de cette opération, nous avons obtenu un seul arbre syntaxique ainsi que sa représentation sémantique illustrée par la figure 5.18.

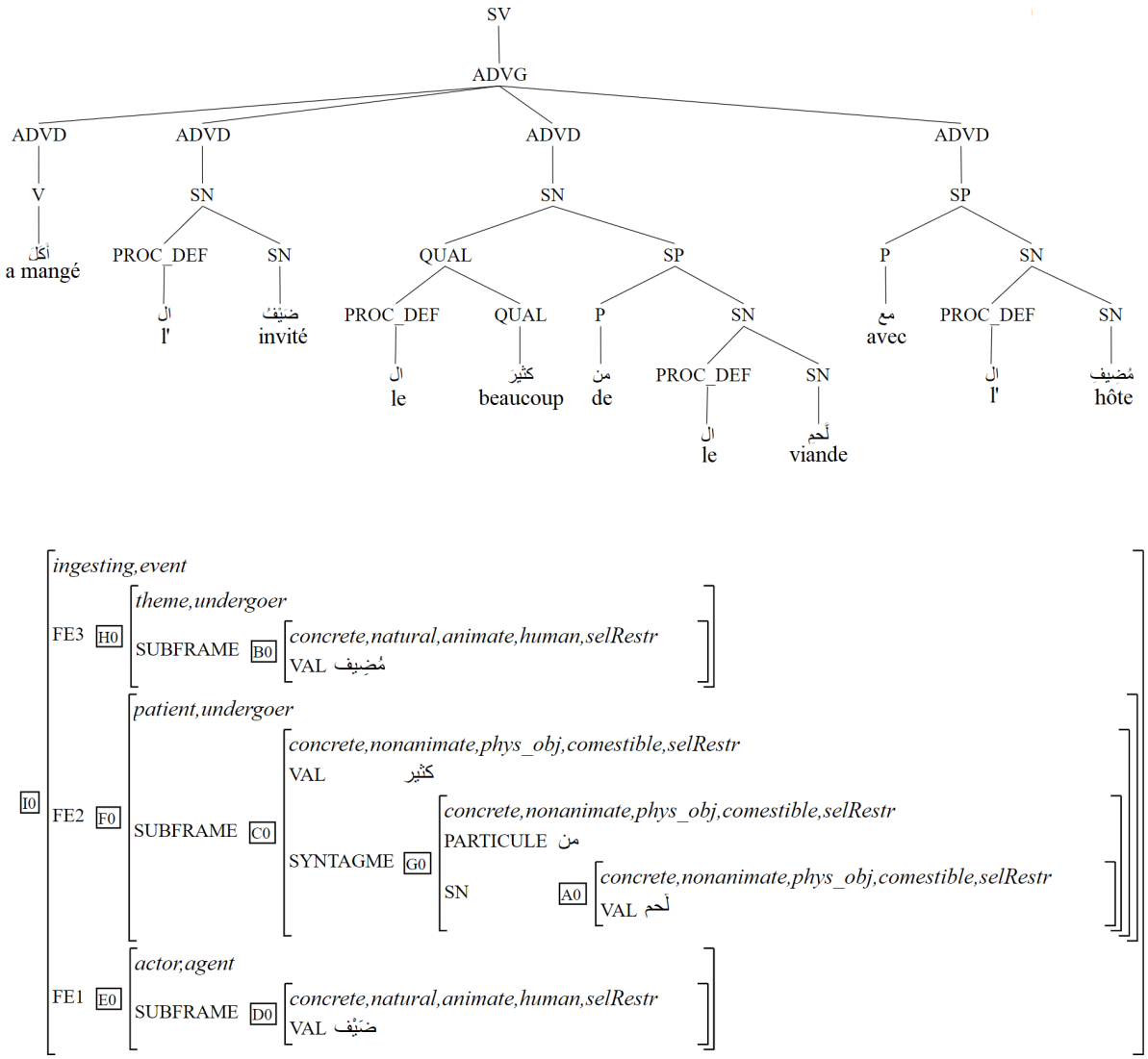


FIGURE 5.18: Résultat de l'analyse syntaxico-sémantique de phrase (L'invité a mangé beaucoup de viande avec l'hôte)

Cet arbre syntaxique correspond au deuxième modèle de la figure 5.17 et il désigne ainsi le sens correct de la phrase. L'ambigüité d'interprétation syntaxique, au niveau de la structure "الكثيرَ من اللَحْم مع المُضِيفِ" (beaucoup de viande avec l'hôte) a été levée lors du processus d'unification des cadres sémantiques correspondants.
En effet, dans le modèle (2) (de la figure 5.17), la structure "الكثيرَ من اللَحْم مع المُضِيفِ" (beaucoup de viande avec l'hôte) représente deux arguments distincts du verbe "أَكَلَ" (manger). Un argument sous forme d'un syntagme nominale "الكثيرَ من اللَّحم" (beaucoup de viande) et un deuxième sous forme d'un syntagme prépositionnel "مع المُضِيفِ" (avec l'hôte). Les types des cadres sémantiques de ces syntagmes sont compatibles avec les contraintes de rôles sémantiques du prédicat. En effet, ce dernier a besoin d'un "Patient" comestible et d'un "Theme" animé. L'unification de leurs cadres est réalisée avec succès. En revanche, dans le premier modèle, la structure correspond à un syntagme nominal composé d'une conjonction entre "اللَّحم" (viande) et "المُضِيفِ" (l'hôte). Le type de cadre du premier syntagme est aliment comestible, tandis que le deuxième est un humain. L'incompatibilité des types a causé l'échec de l'unification de ces cadres. Leur conjonction aurait donné une signification incorrecte à savoir que l'hôte est comestible avec la viande. Néanmoins, nous devons préciser que dans certains cas l'ambiguïté syntaxique ne peut pas être levée lors de l'analyse sémantique. La raison est l'insuffisance d'informations sémantiques indiquant le contexte de la phrase. Par exemple, dans cette phrase : "أبلغنا الدليل" (le guide nous a informés), le sujet "الدليل" (le guide) peut faire référence à un guide touristique ou à un livre / répertoire. Dans le premier cas, il aura le rôle "Agent" alors que dans le second ce sera un "Instrument". Le sens correct ne peut être déterminé qu'en connaissant le contexte global de la phrase. Dans ce cas, l'ambiguïté sémantique ne peut être résolue qu'à un niveau plus élevé à savoir le niveau pragmatique.

5.4 Conclusion

Bien qu'il marque la fin de notre manuscrit, ce chapitre nous ouvre les voies pour progresser et améliorer notre approche. Actuellement, notre grammaire décrit 1074 arbres syntaxiques et 27 cadres sémantiques. Afin de mesurer la couverture et la qualité de cette grammaire, nous avons mis en place trois procédures d'évaluation afin de mesurer :

1. La couverture syntaxique de 1000 phrases grammaticalement correctes extraites d'un livre scolaire arabe. La grammaire a réussi à couvrir correctement 8881 phrases, soit un taux de 88,1% de pourcentage de réussite contre 11,9% de taux d'échec.
2. La non-couverture syntaxique de 250 phrases agrammaticales générées à partir du corpus de test précédent (1000 phrases grammaticalement correctes extraites d'un livre scolaire arabe). Malgré qu'ArabTAG V2.0 n'a pas donné de résultat pour 217 exemples de phrases agrammaticales, nous avons mesuré un taux de 13,02% de couverture de ces phrases incorrectes.
3. Les représentations sémantiques de 460 phrases extraites de la ressource lexicale ArabicVerbNet. Parmi ces 460, seulement 21 phrases n'ont pas été correctement

représentées sémantiquement par notre grammaire.

Les résultats obtenus de ces différentes évaluation sont très encourageants. Ils nous ont apporté les premières estimations sur la qualité et la couverture de notre grammaire. ArabTAG V2.0 couvre les phrases verbales (forme active et passive), les phrases nominales, les différents types des syntagmes nominaux et les syntagmes prépositionnels. Elle traite aussi les différents phénomènes linguistiques arabes tels que la variation des positions des éléments au sein des composants syntaxiques, les compléments supplémentaires, les règles d'accord et les formes agglutinées. Et contrairement aux autres grammaires développées pour l'arabe, elle fournit en plus de la représentation syntaxique de la phrase sa description en cadre sémantique.

Mais d'un autre côté, nous avons pu assimiler et cerner les limites de notre approche. Cela étant dit, remédier à certaines de ces limites est concevable grâce à l'extensibilité de la méta-grammaire d'ArabTAG V2.0. Cette caractéristique permet d'enrichir aisément la grammaire générée et accroitre sa couverture syntaxique et sémantique.

Conclusion générale et perspectives

Dans ce travail de thèse, nous nous sommes intéressés à la problématique de l'analyse syntaxico-sémantique de l'arabe standard moderne. A la différence des autres langues naturelles telles que l'anglais et le français, les ressources numériques utiles pour le traitement de l'arabe demeurent relativement rares. A ce jour, il n'existe pas une grammaire formelle à grande échelle traitant de la syntaxe et la sémantique de l'arabe. Ce constat est appuyé par l'étude des travaux antérieurs que nous avons réalisée. Cette étude nous a permis de relever deux points conséquents. Tout d'abord, la difficulté de construire et entretenir une grammaire à large couverture pour une langue riche et complexe telle que l'arabe. Ensuite, la disponibilité très limitée des ressources telles que les corpus annotés pour l'évaluation des grammaires développées. Dans ce contexte, nous nous sommes orientés vers la construction d'une grammaire permettant une analyse syntaxico-sémantique de l'arabe.
En premier lieu, nous avons choisi un formalisme grammatical adéquat pour représenter ces deux niveaux pour la langue arabe, à savoir les grammaires d'arbres adjoints (TAG). Quant au formalisme de description sémantique, notre choix s'est porté sur les cadres sémantiques. Ensuite, nous nous sommes concentrés sur l'étude des grammaires TAG pour l'arabe qui nous semblaient les plus pertinentes. A la fin de cette étude, nous avons pris comme point de départ ArabTAG qui a été construite manuellement dans le cadre de la thèse de [Ben Fraj, 2010] réalisée au sein du laboratoire RIADI-GDL. Nous avons donc reconsidéré cette grammaire et défini ses limites afin de construire une nouvelle version plus riche et plus fiable.
Dans l'optique de mettre en place une TAG à grande échelle couvrant à la fois les aspects syntaxiques et sémantiques de cette langue nous nous sommes intéressés aux moyens de production semi-automatique de grammaires à savoir les méta-grammaires. A l'aide des formalismes méta-grammaticaux XMG et XMG2 nous avons :

1. Décrit les structures syntaxiques de l'arabe au sein d'une méta-grammaire SynArabTAG. Cette dernière est constituée d'un ensemble de descriptions de fragments d'arbres organisés en famille. Chaque modèle d'arbre de ces familles possède un nœud ancre permettant sa lexicalisation.
2. Défini les représentations sémantiques sous forme de cadres sémantiques au sein d'une méta-grammaire SemArabTAG. A chaque famille d'arbre syntaxique, nous avons décrit son cadre sémantique correspondant.
3. Déterminé une hiérarchie de contraintes des rôles sémantiques et leurs types.
4. Implémenté une interface syntaxe-sémantique reliant les deux niveaux de descriptionSynArabTAG et SemArabTAG.

A l'issue de la compilation de ces descriptions méta-grammaticales, nous avons réussi à générer une nouvelle TAG de l'arabe à portée sémantique, baptisée ArabTAG V2.0. Cette dernière décrit 1074 arbres syntaxiques et 27 cadres sémantiques. Lors de l'analyse syntaxico-sémantique d'une phrase, l'interface syntaxe-sémantique ainsi que la hiérarchie de contrainte implémentées au sein de cette grammaire assurent l'unification des cadres sémantiques correspondants aux constituants de la phrase. À la fin de l'analyse, ArabTAG V2.0 fourni une ou plusieurs représentations syntaxiques et le (ou les) cadre(s) sémantique(s) résultat de la phrase en question.
Durant le développement d'ArabTAG V2.0, nous avons mis en place un processus de validation dans le but de contrôler sa couverture grammaticale. Grâce à un corpus de

phénomènes, que nous avons défini manuellement, nous avons vérifié que la grammaire générée est capable de reconnaître des phrases couvrant des phénomènes linguistiques en arabe tels que la variation des positions des éléments au sein des composants syntaxiques, les compléments optionnels de la phrase, les règles d'accord, les formes agglutinées ainsi que les structures complexes (enchâssées et croisées).La couverture de la grammaire a été également étendue, par rapport à la première version d'ArabTAG (les phrases verbales, phrases nominales, syntagmes nominaux, les syntagmes subordonnés et syntagmes prépositionnels), en ajoutant des arbres élémentaires pour la représentation des compléments tels que les compléments circonstanciels de temps, compléments circonstanciels de lieu et les adverbes.
Pour finir, nous avons appuyé ces résultats en procédant à l'évaluation de notre grammaire en utilisant différents corpus de test. Nous avons à cet effet construit un corpus contenant 1070 phrases annotées syntaxiquement et un autre corpus de 481 phrases annotées syntaxiquement et sémantiquement. Cette évaluation nous a permis d'affirmer la qualité de couverture d'ArabTAG V2.0 et surtout de cerner ses limites. Au terme de cette thèse, nous pouvons dire que nous avons réussi à proposer une nouvelle approche pour analyser syntaxiquement et sémantiquement les phrases de la langue arabe. Nous avons été amenés également à étendre le formalisme méta-grammatical XMG en mettant en place de nouveaux principes traitants des spécificités de la langue arabe.
En outre, nous avons créé des ressources électroniques qui peuvent être utiles pour d'autres applications traitant de la langue arabe à savoir la grammaire ArabTAGV2.0 et les corpus arborés résultats.
En dépit de l'indisponibilité et de la rareté des ressources numériques pour mener à bien nos travaux durant cette thèse, nous avons mis en place notre propre processus d'évaluation. Pour ce faire, nous avons développé un outil pour réaliser l'étiquetage morphosyntaxique et la correspondance sémantique des constituants de la phrase analysée. L'analyse syntaxico-sémantique est réalisée au moyen de l'analyseur TuLiPA-frames. L'alimentation automatique des verbes (du lexique de test saisi) par les rôles sémantiques s'est faite par l'intermédiaire de la ressource lexicale ArabicVerbNet [Mousser, 2010]. Nous avons également exploité cette ressource afin d'assurer une comparaison entre ses phrases exemples annotées et le résultat fourni par notre analyse syntaxico-sémantique. D'ailleurs, nous comptons aussi mettre à disposition nos suites de tests syntaxico-sémantique pour les autres travaux sur l'arabe. [8]
Malgré le travail accompli, nous ne sommes que trop conscients des limites de nos réalisations et du travail qui reste à faire. Les limites que nous avons observé en évaluant notre approche concernent essentiellement le manque de certaines descriptions pour une meilleure gestion des phénomènes linguistiques de l'arabe tel que les règles d'accord en plus du manque de couverture de quelques structures syntaxiques. Ce qui a entrainé des répercussions au niveau de l'analyse syntaxico-sémantique. Néanmoins, l'organisation de notre méta-grammaire en des structures factorisées hiérarchiquement facilite sa maintenance et son extension.
En effet, nous avons analysé les cas d'échecs constatés durant l'évaluation, et nous avons pu délimiter les sources de ces problèmes. Notre perspective à court de terme est de corriger ces erreurs et d'enrichir davantage notre grammaire. Nous envisageons égale-

8. Les ressources éléctroniques que nous avons générés sont accessibles au lien suivant : https ://github.com/Cherifabk

ment d'améliorer et d'enrichir les représentations sémantiques en plus de la hiérarchie de contraintes implémentées au sein de la méta-grammaire. En effet, malgré les premiers résultats encourageants, notre analyse syntaxico-sémantique ne fournit pas de résultats pertinents pour les phrases complexes ayant plus qu'un prédicat. De plus, la ressource lexicale ArabicVerbNet utilisée fournit des contraintes limitées et parfois insuffisantes dans la désambiguïsationde certains cas.
Une autre perspective de nos travaux concerne la méthode d'analyse. Notre approche se base sur l'analyse de textes correctement étiquetés à l'entrée (un "gold" étiquetage). En d'autres termes, l'étiquetage effectué est intrinsèque au processus de l'analyse syntaxico-sémantique. Par conséquent, l'utilisation d'un étiqueteur automatique (POS-tagger) pourrait altérer les résultats. Toutefois, il serait plus pratique d'avoir une chaîne de traitement complètement automatique de notre analyse avec une alimentation plus riche du lexique à partir de dictionnaires électroniques ou encore des bases de données sémantiques telle que FrameNet.
Aussi, notre approche est novatrice dans le sens où elle sépare la description syntaxique de la description sémantique tout en gardant un lien entre ces deux niveaux. En effet, grâce à cette organisation nous pouvons implémenter, un autre format de description sémantique dans la méta-grammaire différent de celui des cadres que nous avons décrit. Il suffirade définir une nouvelle description méta-grammaticale de la dimension sémantique (sémantique plate par exemple) et par la suite la lier à la dimension syntaxique au moyen de l'interface syntaxe-sémantique. Cela permettrait l'émergence de diverses pistes de recherche concernent le calcul sémantique pour les TAG. De plus, cette proposition pourrait servir comme base de comparaison de notre approche puisqu'elle apportera des éléments supplémentaires et de nouveaux horizons pour nos travaux de recherche.
Nous prévoyons donc de poursuivre ces nouvelles pistes de recherche qui ont émergé de nos travaux. Nous aspirons ainsi à mettre en place une TAG à portée sémantique à grande échelle pour la langue arabe et à créer automatiquement un grand corpus arboré riche d'informations syntaxico sémantiques. Ces ressources seront utiles pour les applications du TALN telles que les applications à base d'apprentissage, l'évaluation des analyseurs ou même des grammaires.
Pour conclure nous pouvons dire que la construction des grammaires fut-elle semi-automatisée ou même complètement automatisée reste un travail ardu. Beaucoup de travaux de recherche s'orientent vers des approches à base d'apprentissage automatique pour la construction automatique de grammaires à partir de corpus arborés afin d'éviter les problèmes de couverture, mais encore faut-il disposer de telles ressources. Probablement, comme beaucoup de chercheurs avant nous, nous aurons à nous poser la question de savoir si l'on pourrait bénéficier des résultats des autres approches à base d'apprentissage afin de compléter le manque de couverture de la grammaire ? Pourquoi alors ne pas envisager une approche hybride entre les deux ?

Annexes

.1 Annexe A

Trait	**Description**	**Valeurs possibles**
Cat	Indique la catégorie du nœud au sein de la structure élémentaire	v (verbe), sv (syntagme verbal), sn (syntagme nominal), sp (syntagme prépositionnel), p (préposition), adj (adjectif), advg (adverbe à gauche), advd (adverbe à droite), adv (adverbe), inter (phrase interrogative), pinter (particule d'interrogation), pn (phrase nominale), circ (circonstanciel), subor (outil de liaison pour un syntagme subordonné nominal), subor_p(outil de liaison pour un syntagme subordonné prépositionnel) , pactif (participe actif), ppassif (participe passif), comp (comparatif), qual (qualité similaire), nverbal (nom verbal), proc_v (proclitique verbale), cc (complément circonstanciel), v_e (verbe d'existence), v_c (verbe de certitude), enc (enclitique), proc (proclitique), proc_def (proclitique de détermination), proc_c (proclitique de choix), mushtaq (nom dérivé), exc (phrase exclamative), app (phrase d'appel)

SubCat	Indique la sous-catégorie du nœud au sein de la structure élémentaire	dem (démonstratif), pron (pronom), sn_sub (syntagme subordonné nominal), sn_subp (syntagme subordonné prépositionnel),sn_ann (syntagme d'annexion), nom_prop (nom propre), sn_sem_pac (syntagme quasi-propositionnel reposant sur le participe actif), sn_sem_ppas (syntagme quasi-propositionnel reposant sur le participe passif), sn_sem_comp (syntagme quasi-propositionnel reposant sur un comparatif), sn_sem_qual (syntagme quasi-propositionnel reposant sur la qualité similaire), sn_sem_nv (syntagme quasi-propositionnel reposant sur un nom verbal), sn_adj (syntagme adjectival), sn_def (syntagme défini), sn_com (nom commun), sn_dem (syntagme démonstratif)
Fg	Indique la fonction du nœud au sein de la structure élémentaire	sujet, objet1, objet2, sujet_adj (sujet adjoint)
FgType	Spécifie le type de la fonction du nœud au sein de la structure élémentaire	direct, indirect, interrogative, exclamative, appel
Gen	Spécifie le genre de la structure	m : le genre masculin, f : le genre féminin.
Nombr	Spécifie le nombre de la structure	sg : singulier, dl : dual, plr : pluriel.
Cas	Présente la voyelle de fin du mot	nom : nominatif, acc : accusatif, gen : génétif.
Def	Lorsque le nom est précédé par "ال"	+ : indique que le nom est défini (إسم معرّف) - : indique que le nom est indéfini (إسم غير معرّف)

Pers	Spécifie la personne	De 1 à 13
Voix	présente la voix de conjugaison du verbe, si verbe il y a.	act : actif , pas : passif.
Aspect	Détermine le temps de conjugaison des verbes.	acco : accompli , inacco : inaccompli.
Mode	Le mode du verbe	ind : indicatif , sub : subjonctif, apoc : apocopé, imp : impératif.
Proc(p) /Enc(oclitic)	Présente la possibilité, l'impossibilité ou l'exigence de la présence d'un proclitique (respectivement enclitique) lié au nœud en question	p_e (faux), p_a (vrai) (pour les proclitiques) o_0 (faux), o_1 (vrai) (pour les enclitiques)
Humain	Indique si le nœud au sein de la structure élémentaire est humain ou non	h_0 (faux), h_1 (vrai)
Anime	Indique si le nœud au sein de la structure élémentaire est animé ou non	a_0 (faux), a_1 (vrai)

TABLE 6: L'ensemble de traits d'ArabTAG V2.0

.2 Annexe B

Rôle sémantique	Description du rôle
Actor	L'instigateur de l'évènement
Agent	C'est un acteur qui initie et réalise l'événement intentionnellement. Il existe indépendamment de l'événement.
Asset	C'est une valeur qui désigne un objet concret
Attribute	C'est une propriété d'une ou plusieurs entités qui subit l'événement
Beneficiary	Subit l'événement ou l'état, il peut être avantagé ou désavantagé par l'événement (ou état)
Cause	C'est un acteur qui initie l'événement mais sans aucune intentionnalité ou conscience et qui existe indépendamment de l'événement.
Destination	Location physique vers laquelle quelque chose ou quelqu'un se dirige
Experiencer	C'est un patient qui est conscient de subir l'événement (par exemple les événements de perception)
Extent	C'est une valeur indiquant la quantité de changement mesurable à un participant au cours de l'événement.
Goal	Lieu qui existe indépendamment de l'événement et marque la fin d'une action
Instrument	Subit l'événement mais existe indépendamment de ce dernier. Il est manipulé par un agent, et avec lequel un acte intentionnel est effectué
Location	Un endroit concret
Material	Hérite de patient et source, il est transformé à travers l'événement en une nouvelle entité
Participant	C'est une entité impliquée dans un état ou un événement.
Patient	Subit un changement d'état, d'emplacement ou de condition durant l'événement et existe indépendamment de l'événement
Place	Participant qui représente l'état dans lequel une entité existe
Product	Résultat ou objectif atteint à travers l'événement qui est un objet concret
Recipient	Une destination qui est animée
Source	Lieu qui est le point de départ de l'action. Il existe indépendamment de l'événement
Stimulus	Une cause dans un événement qui suscite une réponse émotionnelle ou psychologique (exemple : événements de perception)
Time	Participant qui indique l'instant ou l'intervalle de temps pendant lequel un état ou un événement a eu lieu.

Theme	Il est au centre d'un événement ou d'un état, n'a pas de contrôle sur la façon dont l'événement se produit et n'est pas structurellement modifié par ce dernier
Topic	Thème caractérisé par le contenu d'information transféré à un autre participant (exemple : événements de communication).
Undergoer	Participant à un état ou un événement mais qui n'est pas un instigateur
VnValue	C'est une place avec une échelle formelle
Oblique	Fait référence au changement d'état, chemin ou "région" subit par un participant (exemple thème) durant l'évènement.
Predicate	C'est un événement ou un état secondaire à l'événement principal (représenté par le verbe prédicat principal)
Proposition	C'est une entité qui participe à l'événement mais sans être l'instigateur.

TABLE 7: Description des rôles sémantiques

Bibliographie

[Abeillé, 1993] Abeillé, A. (1993). *Les nouvelles syntaxes : Grammaires d'unification et analyse du français*. Edition Armand Colin.

[Abeillé, 2002] Abeillé, A. (2002). *Une grammaire électronique du français*. CNRS Editions, Paris.

[Aït-Kaci, 1991] Aït-Kaci, H. (1991). *Warren's Abstract Machine : A Tutorial Reconstruction*. MIT Press, Cambridge, MA, États-Unis.

[Al-Bataineh and Bataineh, 2009] Al-Bataineh, B. and Bataineh, E. (2009). An efficient recursive transition network parser for arabic language. *Lecture Notes in Engineering and Computer Science*, 2177.

[Al-Taani et al., 2012] Al-Taani, A., Msallam, M., and Wedian, S. (2012). A top-down chart parser for analyzing arabic sentences. *IAJIT*, 9.

[Arps and Petitjean, 2018] Arps, D. and Petitjean, S. (2018). A parser for ltag and frame semantics. In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japon. European Language Resource Association.

[Attia, 2008] Attia, M. (2008). *Handling Arabic Morphological and Syntactic Ambiguity within the LFG Framework with a View to Machine Translation.* Thèse doctorale, The University of Manchester, Manchester.

[Baker et al., 1998] Baker, C., Fillmore, C., and Lowe, J. (1998). The berkeley framenet project. In *Proceedings of the 17th International Conference on Computational Linguistics - Volume 1*, COLING '98, pages 86–90, Stroudsburg, PA, États-Unis. Association for Computational Linguistics.

[Becker, 2000] Becker, T. (2000). Patterns in metarules. *In A. Abeille and O. Rambow (Eds.), Tree Adjoining Grammars : formal, computational and linguistic aspects. CSLI publications, Stanford.*

[Belguith et al., 2007] Belguith, L., Aloulou, C., and Hamadou, A. (2007). Maspar : De la segmentation à l'analyse syntaxique de textes arabes. *CÉPADUÈS-Editions, editeur, Revue Information Interaction Intelligence I*, 3 :9–6.

[Ben Fraj, 2010] Ben Fraj, F. (2010). *Un analyseur syntaxique pour les textes en langue arabe à base d'un apprentissage à partir des patrons d'arbres syntaxiques*. Thèse doctorale, ENSI La Manouba, Tunisie.

[Ben Khelil et al., 2018a] Ben Khelil, C., Ben Othmane Zribi, C., Duchier, D., and Parmentier, Y. (2018a). Building a syntactic-semantic interface for asemi-automatically generated TAG for arabic. *Int. Arab J. Inf. Technol.*, 15(3A) :540–549.

[Ben Khelil et al., 2018b] Ben Khelil, C., Ben Othmane Zribi, C., Duchier, D., and Parmentier, Y. (2018b). A semi-automatically generated tag for arabic : Dealing with linguistic phenomena. In *19th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing 2018), Hanoi, Vietnam.*

[Ben Khelil et al., 2016] Ben Khelil, C., Duchier, D., Parmentier, Y., Ben Othmane Zribi, C., and Ben Fraj, F. (2016). Arabtag : from a handcrafted to a semi-automatically generated tag. In *Proceedings of the 12th International Workshop on Tree Adjoining Grammars and Related Formalisms (TAG+12), Heinrich Heine University, Düsseldorf, Allemagne*, pages 18–26.

[Ben Othmane Zribi, 1998] Ben Othmane Zribi, C. (1998). *De la synthèse lexicographique à la détection et à la correction des graphies fautives arabes.* Thèse de doctorat, Université de Paris XI, Orsay.

[Bos, 1995] Bos, J. (1995). Predicate logic unplugged. In *Proceedings of the tenth Amsterdam Colloquium, Amsterdam.*

[Boukedi and Haddar, 2014] Boukedi, S. and Haddar, K. (2014). Hpsg grammar treating of different forms of arabic coordination. *Research in Computing Science*, 86 :25–41.

[Bresnan and Kaplan, 1982] Bresnan, J. and Kaplan, R. M. (1982). Introduction : Grammars as mental representations of language. *The Mental Representation of grammatical relations, MIT Press.*

[Candito, 1996] Candito, M. (1996). A principle-based hierarchical representation of ltags. In *16th International Conference on Computational Linguistics, Proceedings of the Conference, COLING, Center for Sprogteknologi, Copenhague, Danemark*, pages 194–199.

[Candito, 1999] Candito, M. (1999). *Organisation modulaire et paramétrable de grammaires électroniques lexicalisées Application au français et à l'italien.* PhD thesis, Thèse de Doctorat de Linguistique Théorique, Formelle et Automatique, Université Paris 7.

[Chomsky, 1980] Chomsky, N. (1980). A review of bf skinner's verbal behavior. *Readings in philosophy of psychology*, 1 :48–63.

[Christensen et al., 2011] Christensen, J., Mausam, Soderland, S., and Etzioni, O. (2011). An analysis of open information extraction based on semantic role labeling. In *Proceedings of the Sixth International Conference on Knowledge Capture*, K-CAP '11, pages 113–120, New York, NY, États-Unis. ACM.

[Crabbé, 2005] Crabbé, B. (2005). *Représentation informatique de grammaires fortement lexicalisées : Application à la grammaire d'arbres adjoints.* PhD thesis, Thèse de Doctorat, Université Nancy 2.

[Crabbé and Duchier, 2004] Crabbé, B. and Duchier, D. (2004). Metagrammar redux. In *Constraint Solving and Language Processing, First International Workshop, CSLP 2004, Roskilde, Denmark, Revised Selected and Invited Papers*, pages 32–47.

[Crabbé et al., 2013] Crabbé, B., Duchier, D., Gardent, C., Roux, J. L., and Parmentier, Y. (2013). Xmg : extensible metagrammar. *Computational Linguistics*, 39(3) :591–629.

[Diab et al., 2008] Diab, M. T., Moschitti, A., and Pighin, D. (2008). Semantic role labeling systems for arabic using kernel methods. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, Columbus, Ohio, États-Unis*, pages 798–806.

[Dowty et al., 1981] Dowty, D., Wall, R., and Peters, S. (1981). *Introduction to Montague Semantics*, volume 11. Springer Science & Business Media.

[Duchier et al., 2004] Duchier, D., Le Roux, J., and Parmentier, Y. (2004). The metagrammar compiler : An nlp application with a multi-paradigm architecture. In *International Conference on Multiparadigm Programming in Mozart/OZ*, pages 175–187. Springer.

[Duchier and Niehren, 2000] Duchier, D. and Niehren, J. (2000). Dominance constraints with set operators. In *International Conference on Computational Logic*, pages 326–341. Springer.

[Duchier and Thater, 1999] Duchier, D. and Thater, S. (1999). Parsing with tree descriptions : a constraint-based approach. pages 17–32.

[Fader et al., 2011] Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1535–1545. Association for Computational Linguistics.

[Fillmore, 1967] Fillmore, C. J. (1967). The case for case.

[Fillmore, 1982] Fillmore, C. J. (1982). Frame semantics. pages 111–137.

[Gaiffe et al., 2002] Gaiffe, B., Crabbé, B., and Roussanaly, A. (2002). A new metagrammar compiler. In *Proceedings of the Sixth International Workshop on Tree Adjoining Grammar and Related Frameworks, TAG+ 2002, Venise, Italie*, pages 234–241.

[Gardent, 2006] Gardent, C. (2006). Intégration d'une dimension sémantique dans les grammaires d'arbres adjoints. pages 149–158.

[Gardent and Kallmeyer, 2003] Gardent, C. and Kallmeyer, L. (2003). Semantic construction in f-tag. In *EACL 2003, 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 123–130.

[Gerald et al., 1985] Gerald, G., Ewan, K., Geoffrey K., P., and Ivan, S. (1985). *Generalized phrase structure grammar*. Harvard University Press.

[Ghneim et al., 2009] Ghneim, N., Karhely, E., and Sa, W. (2009). First step of building an arabic framenet (afn). In *13th International Business Information Management Conference (13th IBIMA), Arabic Information Processing. Marrakech, Maroc.*

[Habash and Rambow, 2004] Habash, N. and Rambow, O. (2004). Extracting a tree adjoining grammar from the penn arabic treebank. In *In Traitement Automatique du Langage Naturel*, pages 277–284.

[Habash and Roth, 2009] Habash, N. and Roth, R. M. (2009). Catib : The columbia arabic treebank. In *Technical Report CCLS-09-01, Center for Computational Learning Systems, Columbia University.*

[Haddar et al., 2010] Haddar, K., Boukedi, S., and Zalila, I. (2010). Construction of an hpsg grammar for the arabic language and its specification in tdl. *International Journal on Information and Communication Technologies*, 3 :52–64.

[Haddar et al., 2009] Haddar, K., Zalila, I., and Boukedi, S. (2009). A parser generation with the lkb for the arabic relatives. *International Journal of Computing and Information Sciences*, 7 :51–60.

[Hajič et al., 2001] Hajič, J., Hladka, B., and Pajas, P. (2001). The prague dependency treebank : Annotation structure and support. In *Proceedings of the IRCS Workshop on Linguistic Databases*, pages 105–114.

[Hajič et al., 2004] Hajič, J., Smrž, O., Petr, Z., Snaidauf, J., and Beška, E. (2004). Prague arabic dependency treebank : development in data and tools. *Proc. of the NEMLAR Intern. Conf. on Arabic Language Resources and Tools*.

[Hammouda and Haddar, 2017] Hammouda, N. G. and Haddar, K. (2017). Parsing arabic nominal sentences with transducers to annotate corpora. *Computación y Sistemas*, 21(4) :647–656.

[Han et al., 2002] Han, C., Han, N., Ko, E., Palmer, M., and Yi, H. (2002). Penn korean treebank : Development and evaluation. In *Proceedings of the 16th Pacific Asia Conference on Language, Information and Computation. The Korean Society for Language and Information*, pages 69–78.

[Harman, 1963] Harman, G. (1963). Generative grammars without transformation rules : A defense of phrase structure. *Language*, 39 :597–616.

[Joshi, 1987] Joshi, A. K. (1987). An introduction to tree adjoining grammars. *Mathematics of language*, 1 :87–115.

[Joshi et al., 1975] Joshi, A. K., Levy, L. S., and Takahashi, M. (1975). Tree adjunct grammars. *Journal of computer and system sciences*, 10(1) :136–163.

[Joshi and Schabes, 1992] Joshi, A. K. and Schabes, Y. (1992). Tree-adjoining grammars and lexicalized grammars. *In Maurice Nivat and Andreas Podelski (editors), Tree Automata and Languages. Elsevier Science*.

[Joshi and Vijay-Shanker, 2001] Joshi, A. K. and Vijay-Shanker, K. (2001). Compositional semantics with lexicalized tree-adjoining grammar (ltag) : How much underspecification is necessary ? In *Computing Meaning*, pages 147–163. Springer.

[Kallmeyer and Osswald, 2013] Kallmeyer, L. and Osswald, R. (2013). Syntax-driven semantic frame composition in lexicalized tree adjoining grammars. *Journal of Language Modelling Vol*, 1(2) :267–330.

[Kasper, 2008] Kasper, S. (2008). *A comparison of "thematic role" theories*. Thèse doctorale.

[Kingsbury and Palmer, 2003] Kingsbury, P. and Palmer, M. (2003). Propbank : The next level of treebank. In *Proceedings of Treebanks and Lexical Theories*, Växjö, Sweden.

[Kipper et al., 2008] Kipper, K., Korhonen, A., Ryant, N., and Palmer, M. (2008). A large-scale classification of english verbs. *Language Resources and Evaluation*, 42(1) :21–40.

[Kouloughli, 1992] Kouloughli, D. (1992). *La grammaire Arabe pour tous*. Press Pocket.

[Lecomte, 2004] Lecomte, A. (2004). *Méthodes pour le Traitement Automatique des Langues*. M1 Ingénierie de la Communication Personne -Système.

[Levin, 1993] Levin, B. (1993). *English Verb Classes and Alternations :˜A Preliminary Investigation*. University of Chicago Press.

[Lichte, 2007] Lichte, T. (2007). An mctag with tuples for coherent constructions in german. In *Proceedings of the 12th Conference on Formal Grammar*.

[Lichte and Petitjean, 2015] Lichte, T. and Petitjean, S. (2015). Implementing semantic frames as typed feature structures with xmg. *Journal of Language Modelling*, 3 :185–228.

[Liu and Gildea, 2010] Liu, D. and Gildea, D. (2010). Semantic role features for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 716–724, Stroudsburg, PA, États-Unis. Association for Computational Linguistics.

[Loukam and Laskri, 2008] Loukam, M. and Laskri, M. T. (2008). Pharas : Une plateforme d'analyse basée sur le formalisme hpsg pour l'arabe standard : Développements récents et perspectives.

[Maamouri and Bies, 2004] Maamouri, M. and Bies, A. (2004). Developing an arabic treebank : Methods, guidelines, procedures, and tools. In *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, Semitic '04, pages 2–9, Stroudsburg, PA, États-Unis. Association for Computational Linguistics.

[Maamouri et al., 004a] Maamouri, M., Bies, A., Buckwalter, T., and Jin, H. (2004a). Arabic treebank : Part 2 v 2.0. linguistic data consortium, catalog number ldc2004t02, isbn : 1-58563-282-1.

[Maamouri et al., 004b] Maamouri, M., Bies, A., Buckwalter, T., and Jin, H. (2004b). Arabic treebank : Part 3 v 1.0. linguistic data consortium, catalog number ldc2004t11, isbn : 1-58563-298-8.

[Maamouri et al., 2005] Maamouri, M., Bies, A., Buckwalter, T., Jin, H., and Mekki, W. (2005). Arabic treebank : Part 4 v 1.0 linguistic data consortium, catalog number ldc2005t30.

[Maamouri et al., 2003] Maamouri, M., Bies, A., Jin, H., and Buckwalter, T. (2003). Arabic treebank : Part 1 v 2.0. linguistic data consortium, catalog number ldc2003t06, isbn : 1-58563-261-9.

[Maamouri et al., 2010] Maamouri, M., Graff, D., Bouziri, B., Krouna, S., Bies, A., and Kulick, S. (2010). Ldc standard arabic morphological analyzer (sama) version 3.1 ldc2010l01.

[Maqsud et al., 2014] Maqsud, U., Arnold, S., Hülfenhaus, M., and Akbik, A. (2014). Nerdle : Topic-specific question answering using wikia seeds. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics : System Demonstrations*, pages 81–85, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

[Marcus et al., 1993] Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a large annotated corpus of english : The penn treebank. *Comput. Linguist.*, 19(2) :313–330.

[Masuichi et al., 2003] Masuichi, H., Ohkuma, T., Yoshimura, H., and Harada, Y. (2003). Japanese parser on the basis of the lexical-functional grammar formalism and its evaluation. In *The 17th Pacific Asia Conference on Language, Information and Computation (PACLIC 17)*, pages 298–309.

[Meguehout et al., 2017] Meguehout, H., Bouhadada, T., and Laskri, M. T. (2017). Semantic role labeling for arabic language using case-based reasoning approach. *I. J. Speech Technology*, 20 :363–372.

[Montague, 1970] Montague, R. (1970). Universal grammar. *Theoria*, 36(3) :373–398.

[Montague, 1974a] Montague, R. (1974a). English as a formal language. In *Selected papers of Richard Montague*, pages 188–221.

[Montague, 1974b] Montague, R. (1974b). Formal philosophy. *Selected Papers of Richard Montague*.

[Mousser, 2010] Mousser, J. (2010). A large coverage verb taxonomy for arabic. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, La Valette, Malte*.

[Mousser, 2011] Mousser, J. (2011). Classifying arabic verbs using sibling classes. In *Proceedings of the Ninth International Conference on Computational Semantics, IWCS 2011, Oxford, Royaume-Uni*.

[Muskens and Krahmer, 1998] Muskens, R. and Krahmer, E. (1998). Description theory, ltags and underspecified semantics. In *Proceedings of the Fourth International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+ 4)*, pages 112–115.

[Notari, 2010] Notari, C. (2010). *Chomsky et l'ordinateur : approche critique d'une théorie linguistique*. Collection INTERLANGUES linguistique et didactique sous la direction de Wilfrid Rotgé.

[Othman et al., 2003] Othman, E., Shaalan, K., and Rafea, A. (2003). A chart parser for analyzing modern standard arabic sentence. The MT Summit IX Workshop on Machine Translation for Semitic Languages : Issues and Approaches. Nouvelle-Orléans, Louisiane, États-Unis.

[Palmer et al., 2008] Palmer, M., Babko-Malaya, O., Bies, A., Diab, M., Maamouri, M., Mansouri, A., and Zaghouani, W. (2008). A pilot Arabic Propbank. In *LREC 2008*.

[Palmer et al., 2005] Palmer, M., Gildea, D., and Kingsbury, P. (2005). The proposition bank : An annotated corpus of semantic roles. *Computational Linguistics*, 31(1) :71–106.

[Parmentier, 2007] Parmentier, Y. (2007). *SemTAG : une plate-forme pour le calcul sémantique à partir de Grammaires d'Arbres Adjoints*. Thèse doctorale, Université Henri Poincaré - Nancy I.

[Parmentier et al., 2008] Parmentier, Y., Kallmeyer, L., Lichte, T., Maier, W., and Dellert, J. (2008). Tulipa : A syntax-semantics parsing environment for mildly context-sensitive formalisms. In *9th International Workshop on Tree-Adjoining Grammar and Related Formalisms (TAG+9)*, pages 121–128, Tübingen, Allemagne.

[Pereira and Warren, 1980] Pereira, F. C. and Warren, D. H. (1980). Definite clause grammars for language analysis—a survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence*, 13(3) :231–278.

[Perrier, 2000] Perrier, G. (2000). Interaction grammars. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 2*, COLING '00, pages 600–606, Stroudsburg, PA, États-Unis. Association for Computational Linguistics.

[Petitjean, 2014] Petitjean, S. (2014). *Modular generation of formal grammars*. Thèse doctorale, Université d'Orléans.

[Petitjean et al., 2016] Petitjean, S., Duchier, D., and Parmentier, Y. (2016). Xmg2 : Describing description languages. In Amblard, M., de Groote, P., Pogodalla, S., and

Rétoré, C., editors, *Logical Aspects of Computational Linguistics (LACL 2016)*, volume 10054 of *Lecture Notes in Computer Science*, pages 255–272, Nancy, France. Springer-Verlag.

[Piattelli-Palmarini, 1980] Piattelli-Palmarini, M. (1980). *Language and Learning : The Debate Between Jean Piaget and Noam Chomsky*. Harvard University Press.

[Pizzato and Mollá, 2008] Pizzato, L. A. and Mollá, D. (2008). Indexing on semantic roles for question answering. In *Coling 2008 : Proceedings of the 2nd workshop on Information Retrieval for Question Answering*, pages 74–81, Manchester, Royaume Uni. Coling 2008 Organizing Committee.

[Pollard and Sag, 1994] Pollard, C. and Sag, I. A. (1994). Head-driven phrase structure grammar.

[Prolo, 2002] Prolo, C. A. (2002). Generating the xtag english grammar using metarules. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, États-Unis. Association for Computational Linguistics.

[Ratnaparkhi et al., 1994] Ratnaparkhi, A., Roukos, S., and Ward, T. (1994). A maximum entropy model for parsing. In *In Proceedings of the International Conference on Spoken Language Processing, Yokohama, Japon*, pages 803–806.

[Ristad, 1987] Ristad, E. S. (1987). Gpsg-recognition is np-hard. *Linguistic Inquiry*, 18(3) :530–536.

[Rogers and Vijay-Shanker, 1992] Rogers, J. and Vijay-Shanker, K. (1992). Reasoning with descriptions of trees. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pages 72–80, Newark, Delaware, États-Unis. Association for Computational Linguistics.

[Romero and Kallmeyer, 2005] Romero, M. and Kallmeyer, L. (2005). Scope and situation binding in ltag using semantic unification.

[Ruwet, 1968] Ruwet, N. (1968). *Introduction à la grammaire générative*.

[Schabes and Joshi, 1990] Schabes, Y. and Joshi, A. K. (1990). Parsing with lexicalized tree adjoining grammar.

[Sgall et al., 1986] Sgall, P., Hajicová, E., and Panevová, J. (1986). *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. D. Reidel et Academia.

[Shieber and Schabes, 1990] Shieber, S. M. and Schabes, Y. (1990). Synchronous tree-adjoining grammars. In *Proceedings of the 13th Conference on Computational Linguistics - Volume 3*, COLING '90, pages 253–258, Stroudsburg, PA, États-Unis. Association for Computational Linguistics.

[Smrž, 2007] Smrž, O. (2007). Elixirfm : Implementation of functional arabic morphology. In *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages : Common Issues and Resources*, Semitic '07, pages 1–8, Stroudsburg, PA, États-Unis. Association for Computational Linguistics.

[Thomasset and De La Clergerie, 2005] Thomasset, F. and De La Clergerie, r. (2005). Comment obtenir plus des méta-grammaires. volume 5. Proceedings of TALN.

[Van Riemsdijk, 1982] Van Riemsdijk, H. (1982). *The Generative Enterprise - A Discussion with Noam Chomsky (with Riny Huybregts)*. Dordrecht : Foris.

[Vapnik, 1998] Vapnik, V. N. (1998). *Statistical Learning Theory*. JohnWiley and Sons.

[Vijay-Shanker and Joshi, 1991] Vijay-Shanker, K. and Joshi, A. (1991). Unification-based tree adjoining grammars. *Technical Reports (CIS)*.

[Vijay-Shanker and Schabes, 1992] Vijay-Shanker, K. and Schabes, Y. (1992). Structure sharing in lexicalized tree-adjoining grammars. In *Proceedings of the 14th Conference on Computational Linguistics - Volume 1*, COLING '92, pages 205–211.

[Villemonte De la Clergerie, 2005] Villemonte De la Clergerie, E. (2005). Dyalog : a tabular logic programming based environment for nlp. In *Proceedings of CSLP'05*.

[Weir, 1988] Weir, D. J. (1988). *Characterizing Mildly Context-sensitive Grammar Formalisms*. PhD thesis.

[Xia, 2001] Xia, F. (2001). *Automatic Grammar Generation from Two Different Perspectives*. Thèse doctorale.

[XTAG, 2001] XTAG, R. G. (2001). A lexicalized tree adjoining grammar for english. *Technical Report IRCS-01-03, IRCS, University of Pennsylvania*.

[Xue et al., 2002] Xue, N., Chiou, F.-D., and Palmer, M. (2002). Building a large-scale annotated chinese corpus. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pages 1–8.

[Yngve, 1960] Yngve, V. H. (1960). A model and an hypothesis for language structure. *Proceedings of the American Philosophical Society*, 104(5) :444–466.

Cherifa BEN KHELIL

Construction semi-automatique d'une grammaire d'arbres adjoints pour l'analyse syntaxico-sémantique de l'arabe

Résumé : Cette thèse traite de la description formelle et du développement d'une grammaire électronique de la langue arabe. Ce travail est un prérequis à la création d'outils de traitement automatique de l'arabe. Cette langue présente de nombreux défis pour un traitement automatique. En effet l'ordre de mots en arabe est relativement libre, la morphologie y est riche et les diacritiques sont omis dans les textes écrits. Bien que plusieurs travaux de recherche aient abordé certaines de ces problématiques, les ressources électroniques utiles pour le traitement de l'arabe demeurent relativement rares ou encore peu disponibles. Dans ce travail de thèse, nous nous sommes intéressés à la représentation de la syntaxe (ordre des mots) et du sens de l'arabe standard moderne. Comme système formel de représentation de la langue, nous avons choisi le formalisme des grammaires d'arbres adjoints (Tree Adjoining Grammar). Nous avons ainsi proposé une grammaire d'arbres adjoints électronique de l'arabe nommée « ArabTAG V2.0 ». Cette ressource réutilise en partie la modélisation préexistante dans la grammaire définie manuellement «ArabTAG » et l'intègre à une représentation abstraite appelée méta-grammaire. L'expert linguiste peut ainsi décrire la syntaxe et sémantique de la langue avec des outils d'abstraction facilitant la maintenance et l'extension de la grammaire. La grammaire ainsi décrite compte 1074 règles syntaxiques (non lexicalisées) et 27 cadres sémantiques (relations prédicatives). Cette ressource a été évaluée en analysant un corpus issu d'extraits d'un manuel scolaire d'apprentissage de l'arabe.

Mots clés : Grammaire d'arbres adjoints, méta-grammaire, syntaxe, sémantique, interface syntaxe/sémantique, cadre sémantique, langue arabe.

Semi-automatic construction of a Tree-adjoining grammar for syntactic-semantic analysis of Arabic

Abstract : This thesis deals with the formal description and development of an electronic grammar of Arabic language. This work is a prerequisite for the creation of automatic Arabic processing tools. This language presents many challenges for automatic processing. Indeed the order of words in Arabic is relatively free, the morphology is rich and the diacritics are omitted in written texts. Although several research studies have addressed some of these issues, electronic resources useful for the processing of Arabic remain relatively rare or not widely available. In this thesis work, we are interested in the representation of syntax (word order) and the meaning of modern standard Arabic. As a formal system of language representation, we chose the formalism of Tree Adjoining Grammar. Thus we proposed an electronic adjoint tree grammar of Arabic named "ArabTAGV2.0". This resource partially reuses the pre-existing modeling in the manually defined grammar "ArabTAG" and integrates it into an abstract representation called meta-grammar. The linguistic expert can thus describe the syntax and semantics of the language with abstraction tools facilitating the maintenance and extension of the grammar. The new described grammar has 1074 syntactical rules (not lexicalized) and 27 semantic frameworks (predicative relations). This resource was evaluated by analyzing a corpus from excerpts of an Arabic textbook.

Keywords : Tree-adjoining grammar, meta-grammar, syntax, semantics, syntax / semantic interface, semantic frames, Arabic language.

LIFO - Laboratoire d'Informatique Fondamentale d'Orléans / RIADI - Laboratoire de Recherche en génie loglciel, Applications distribuées, systèmes Décisionnels et Imagerie intelligente

Bâtiment 3IA, rue Léonard de Vinci, B.P. 6759 45067 ORLEANS cedex 2, FRANCE / Campus Universitaire de la Manouba, 2010 Manouba, Tunisie